

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТВЕРСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

*На правах рукописи*



ГОНЧАРОВ АНДРЕЙ АЛЕКСАНДРОВИЧ

**ОПТИМИЗАЦИЯ ПРОЦЕССА ОБРАБОТКИ ИНФОРМАЦИИ ПРИ  
СЕРТИФИКАЦИИ ПРОДУКЦИОННЫХ ЭКСПЕРТНЫХ СИСТЕМ  
ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ**

Специальность 05.13.01 – «Системный анализ, управление и обработка  
информации (в промышленности)»

**Диссертация  
на соискание ученой степени кандидата  
технических наук**

Научный руководитель:  
доктор технических наук, профессор  
Семенов Николай Александрович

Тверь – 2019 г.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
Глава 1. Анализ процесса сертификации экспертных ИС встраиваемого ПО АСУ ТП.....	17
1.1. Особенности встраиваемого ПО АСУ ТП.....	17
1.2. Промышленные производственные экспертные ИС .....	20
1.3. Формирование требований к сертификации промышленных производственных экспертных ИС .....	27
1.4. Особенности логического вывода производственных экспертных ИС промышленного назначения. Постановка задачи исследования .....	28
1.5. Выводы по главе.....	29
Глава 2. Обоснование методов решения задачи повышения скорости логического вывода производственных экспертных ИС промышленного назначения.....	31
2.1. Известные подходы повышения скорости логического вывода промышленных производственных экспертных ИС .....	31
2.1.1. Фильтры .....	31
2.1.2. Фреймы .....	31
2.1.3. Дуальные экспертные системы .....	31
2.1.4. Нейронные сети.....	32
2.1.5. Генетические алгоритмы.....	32
2.1.6. Rete-алгоритм .....	33
2.2. Подтверждение неприменимости известных подходов повышения скорости логического вывода промышленных производственных экспертных ИС при сертификации встраиваемого ПО АСУ ТП.....	33
2.3. Алгоритмы решения задачи повышения скорости вычислительных операций в производственных экспертных ИС с использованием аспектно- ориентированного подхода .....	34
2.3.1. Особенности аспектно-ориентированного подхода.....	34
2.3.2. Одноитерационный алгоритм извлечения аспектов из последовательности фактов .....	45
2.3.3. Рекурсивный алгоритм извлечения аспектов из последовательности фактов.....	47
2.4. Выводы по главе.....	49

Глава 3. АНАЛИЗ КРИТЕРИЕВ ПОВЫШЕНИЯ СКОРОСТИ ВЫЧИСЛИТЕЛЬНЫХ ОПЕРАЦИЙ В ПРОМЫШЛЕННЫХ ПРОДУКЦИОННЫХ ЭКСПЕРТНЫХ ИС .....	51
3.1. Исходные данные .....	51
3.2. Анализ критериев повышения скорости вычислительных операций в продукционных экспертных ИС .....	54
3.3. Результаты применения разработанных алгоритмов решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС .....	55
3.3.1. Одноитерационный алгоритм извлечения аспектов из последовательности фактов .....	55
3.3.2. Рекурсивный алгоритм извлечения аспектов из последовательности фактов .....	57
3.4. Использование МАИ для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в базе знаний экспертной ИС .....	60
3.4.1. Структуризация задачи в виде иерархической структуры с несколькими уровнями .....	63
3.4.2. Выполнение попарных сравнений критериев .....	64
3.4.3. Вычисление коэффициентов важности для критериев .....	65
3.4.4. Выполнение попарных сравнений альтернатив .....	66
3.4.5. Вычисление коэффициентов важности для альтернатив .....	66
3.4.6. Определение наилучшей альтернативы .....	68
3.5. Выводы по главе .....	72
Глава 4. ПРОТОТИП ПОДСИСТЕМЫ ИНТЕРПРЕТАЦИИ ПРОДУКЦИОННЫХ ПРАВИЛ ЭКСПЕРТНЫХ ИС С ПРИМЕНЕНИЕМ АО-ПОДХОДА .....	73
4.1. Изменения модели экспертной ИС в части интерпретации продукционных правил, базирующиеся на применении АО-подхода .....	73
4.2. Описание реализации прототипа подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода .....	75
4.3. Выводы по главе .....	77
ЗАКЛЮЧЕНИЕ .....	78
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ .....	89
СПИСОК ЛИТЕРАТУРЫ .....	90

ПРИЛОЖЕНИЕ А .....	101
ПРИЛОЖЕНИЕ Б.....	126
ПРИЛОЖЕНИЕ В .....	127

## ВВЕДЕНИЕ

### Актуальность работы

Развитие промышленных предприятий в настоящее время невозможно без применения автоматизированных систем управления производственными и технологическими процессами (АСУ ТП). Наибольший экономический успех сегодня сопутствует только тем предприятиям, которые активно используют современные информационные технологии [57].

Задачу автоматизации образует автоматическое управление совокупность технических процессов предприятия [10].

Встраиваемое (встроенное) программное обеспечение (ПО) АСУ ТП, помимо выполнения целевых функций, должно противостоять следующим угрозам обрабатываемой информации:

- неправомерный доступ
- уничтожение
- модифицирование
- блокирование
- копирование
- предоставление
- распространение
- иные неправомерные действия, в том числе деструктивные информационные воздействия (компьютерные атаки), следствием которых может стать нарушение функционирования АСУ ТП [63].

Примером ПО АСУ ТП являются системы обнаружения вторжений (СОВ).

Авторы и исследователи Ю.А. Кудеяров, А.В. Шестаков, А.А. Дудыкин, А.Н. Паньков и др. выделяют следующие характерные особенности встраиваемого ПО АСУ ТП:

- значительный объем обрабатываемых информационных потоков, сложность и многогранность процессов обработки и передачи информации;

- многообразие функциональных возможностей;
- критическая важность бесперебойного поддержания работоспособности;
- болезненная, с точки зрения совместимости и технической поддержки, реакция на любые изменения в аппаратную или программную составляющие;
- необходимость особенного внимания к обеспечению триады свойств безопасности информационных ресурсов – конфиденциальности, целостности и доступности.

Аккумуляция эмпирических знаний о принимаемых при управлении технологическими процессами решениях привело к использованию в составе встраиваемого ПО АСУ ТП производственных экспертных информационных систем (ИС) промышленного назначения [96]. Производственные экспертные ИС также называются системами на основе правил. Производственная модель чаще всего применяется в промышленных экспертных ИС ([47], [59]).

Именно встраиваемое ПО АСУ ТП является самым уязвимым техническим звеном любой системы. Наличие уязвимостей служит главной причиной возможных сетевых атак и вирусных эпидемий [55]. Уязвимости причинить значительный экономический ущерб предприятию или нанести серьезный удар по авторитету [76].

В силу представленных выше характерных особенностей встраиваемого ПО АСУ ТП, согласно [63], а также исходя из результатов работы специалистов компании Positive Technologies [95], в которой отмечается тот факт, что количество уязвимостей АСУ ТП постоянно увеличивается и в 2019 году увеличилось на 30% по сравнению с предыдущим годом, возникает необходимость осуществления контроля, которая находит своё отражение в процедуре сертификации промышленных ИС.

Сертификация средств защиты информации встраиваемого ПО АСУ ТП производится в соответствии с утвержденным приказом № 55 ФСТЭК России от 3

апреля 2018 г. «Положением о системе сертификации средств защиты информации» [66].

Кроме сертификации можно встретить другие определения различных процедур оценки соответствия, а именно испытания, аттестация, тестирование, аудит, анализ рисков [56].

При сертификации встраиваемого ПО АСУ ТП на первый план выступают сроки проведения всех необходимых проверок.

Важность сосредоточения на уменьшении продолжительности сертификационных испытаний обусловлена невозможностью снижения конкурентоспособности предприятия, а также невозможностью приостановки эффективного процесса изготовления определенного продукта и его реализации предприятием [50].

Промышленность России представлена множеством отраслей и предприятий. Сложившееся обвальное нарастание массы самой разнообразной информации о технологиях промышленности, знаний и метазнаний о промышленных процессах наделяет современные экспертные ИС встраиваемого ПО АСУ ТП большим количеством производственных правил.

Также важно учитывать, исходя из того, что одной из моделей жизненного цикла, которая успешно используется во многих проектах экспертных ИС, является линейная модель [29], БЗ экспертных ИС часто дополняются производственными правилами.

Авторы Валетов В.А., Орлова А.А., Третьяков С.Д. и др. рассматривали проблемы экспертных ИС и в качестве одного из актуальных недостатков производственных экспертных ИС выделяют меньшую эффективность процесса логического вывода, по сравнению с другими моделями представления знаний (МПЗ). Причину этого можно найти в том, что при логическом выводе требуется значительное количество времени на перебор и непроизводительные проверки применимости правил [8].

С одной стороны, отсутствие оперативного устранения со стороны разработчиков уязвимостей, является частой и характерной особенностью

встраиваемого ПО АСУ ТП, что, определенно, увеличивает продолжительность сертификационных испытаний [19]. С другой стороны, на продолжительность сертификационных испытаний влияет количество правил в базе знаний (БЗ) сертифицируемых производственных экспертных ИС.

Приемлемый уровень качества встраиваемого ПО – важнейший фактор конкурентоспособности продукции промышленного предприятия ([38], [67]).

Представленные требования к встраиваемому ПО АСУ ТП показывают важность и актуальность проблемы повышения эффективности процесса логического вывода в производственных экспертных ИС.

### **Объект исследования**

Методы и средства обработки информации в экспертных ИС при сертификации встраиваемого ПО АСУ ТП.

### **Предмет исследования**

Процесс логического вывода производственных экспертных ИС встраиваемого ПО АСУ ТП.

### **Цель работы**

Целью диссертационной работы является повышение эффективности процесса логического вывода в производственных экспертных ИС при сертификации встраиваемого ПО АСУ ТП.

Исходя из цели, в диссертации сформулированы и решены следующие **задачи исследования:**

1. Формирование особенностей сертификации производственных экспертных ИС встраиваемого ПО АСУ ТП.
2. Анализ применяемых в настоящее время методов повышения скорости вычислительных операций в производственных экспертных ИС встраиваемого ПО АСУ ТП.
3. Разработка алгоритмов решения задачи повышения скорости вычислительных операций в производственных экспертных ИС

встраиваемого ПО АСУ ТП с использованием аспектно-ориентированного (АО) подхода.

4. Использование метода анализа иерархий (МАИ) для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в базе знаний экспертной ИС.
5. Описание изменений модели экспертной ИС в части интерпретации продукционных правил, базирующихся на применении АО-подхода, при сертификации встраиваемого ПО АСУ ТП.
6. Разработка прототипа подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода.

### **Методологическая база исследования**

В диссертационной работе использованы:

- методы теории множеств
- методы системного анализа
- методы математической статистики
- методы разработки программного обеспечения ЭВМ.

Вместе с тем среди используемых методов, следует отметить АО-подход, который для повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП ранее не применялся.

### **Теоретическая основа исследования**

Теоретическую основу диссертационной работы составляют труды отечественных и зарубежных ученых и исследователей:

- в области искусственного интеллекта: Е. Alpaydin, М. Sasikumar, R. Chandrasekar, S. Muthu Raman, S. Ramani, А.Б. Барский, А.Л. Каткова, А.П. Частиков, В.Б. Новосельцев, В.В. Курейчик, В.М. Курейчик, В.Ф. Хорошевский, Г. Райли, Д. Джарратано, Д. Рутковская, Д.Л. Белов, Д.Ф. Люгер, Е.Б. Кисель, Е.И. Горожанина, Ж.-Л. Лорьер, И.Б. Фоминых, И.М.

- Братко, К.Р. Червинская, Л. Рутковский, Л.А. Гладков, М. Пилиньский, М.Б. Беркинблит, М.Д. Шапот, Н. Нильсон, Н.А. Семенов, П. Джексон, Р. Каллан, С.В. Аксенов, С.И. Золотов, Т.А. Гаврилова, Э.В. Попов и др.
- в области системного анализа, теории множеств и математической статистики: E.W. Dijkstra, G. Kiczales, М.М. Aksit, Т. Elrad, А.А. Воронов, А.А. Денисов, А.А. Орлова, А.Б. Хуторецкий, А.В. Антонов, А.Н. Катулев, А.П. Афанасьев, Б.А. Гладких, Б.В. Палюх, В.А. Валетов, В.Н. Волкова, Д.М. Жилин, К.Н. Лебедев, Н.А. Северцев, Н.Г. Зайцев, Н.М. Капустин, С.Д. Третьяков, С.М. Дзюба, Т.Я. Данелян, Т.Я. Лазарева, Ю.В. Сидельников, Ю.С. Усынин, Ю.Ф. Мартемьянов и др.
  - в области разработки программного обеспечения ЭВМ и сертификации: A. Theedom, E. Gamma, H. Yang, J. Vlissides, M. Yener, R. Helm, R. Johnson, R. Pawlak, А.А. Демирский, А.А. Дудыкин, А.В. Барабанов, А.В. Шестаков, А.Г. Сергеев, А.Н. Паньков, А.С. Марков, А.Ф. Иголкин, А.Ф. Кулаков, Б.И. Герасимов, Б.М. Позднеев, В.А. Колпаков, В.В. Домарев, В.В. Липаев, В.И. Колчков, В.Л. Цирлов, В.М. Панорядов, Г.Г. Панкова, Д.А. Скрипник, Д.Н. Шевцов, Е.Б. Герасимова, Е.Н. Николенко, И.М. Лифиц, М.Я. Марусина, О.В. Новикова, С.А. Вологжанина, С.А. Матяш, С.Д. Поляков, С.Л. Котов, Ю.А. Кудеяров, Ю.В. Димов и др.

Кроме того, в работе были использованы Государственные стандарты Российской Федерации в области информационной безопасности и информационных технологий.

### **Научная новизна**

Научная новизна диссертационного исследования заключается в следующем:

1. Предложенные одноитерационный и рекурсивный алгоритмы решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП отличаются применением АО-подхода.

2. Обоснование выбора наиболее оптимального алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в БЗ экспертной ИС встраиваемого ПО АСУ ТП отличается использованием МАИ.
3. Изменение модели экспертной ИС встраиваемого ПО АСУ ТП в части интерпретации продукционных правил отличается применением одноитерационного и рекурсивного алгоритмов решения задачи повышения скорости логического вывода в продукционных экспертных ИС.
4. Прототип подсистемы интерпретации продукционных правил экспертной ИС, применяемый при сертификации встраиваемого ПО АСУ ТП, отличается применением АО-подхода.

***Основные положения, выносимые на защиту:***

- 1. Алгоритмы решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП с использованием АО-подхода.**

Соответствует пункту 13 паспорта специальности: Методы получения, анализа и обработки экспертной информации.

- 2. Использование МАИ для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в БЗ экспертной ИС встраиваемого ПО АСУ ТП.**

Соответствует пункту 10 паспорта специальности: Методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических системах.

- 3. Изменение модели экспертной ИС встраиваемого ПО АСУ ТП в части интерпретации продукционных правил, базирующееся на применении АО-подхода.**

Соответствует пункту 4 паспорта специальности: Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации.

#### **4. Прототип подсистемы интерпретации продукционных правил экспертной ИС, применяемый при сертификации встраиваемого ПО АСУ ТП.**

Соответствует пункту 5 паспорта специальности: Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации.

#### **Достоверность и обоснованность результатов**

Обоснованность научных положений, выводов и практических рекомендаций, полученных в диссертационной работе, обеспечивается обоснованностью применяемых методов исследования, доказательством сформулированных утверждений, близостью и согласованностью представленных результатов экспериментальных исследований с теоретическими выводами.

#### **Практическая значимость**

Практическая значимость работы заключается в том, что рассматриваемая актуальная проблема недостаточной эффективности экспертных ИС, представленных большим множеством продукционных правил, может быть решена путём использования АО-подхода. АО-подход позволяет выполнить реорганизацию продукционных правил, отделив сквозные (пересекающиеся) факты.

МАИ показал, что с увеличением количества продукционных правил в БЗ экспертной ИС предпочтительнее использовать рекурсивный алгоритм извлечения аспектов из последовательности фактов.

## **Внедрение результатов работы**

Автором разработаны алгоритмы, базирующиеся на АО-подходе, которые для повышения скорости вычислительных операций в производственных экспертных ИС встраиваемого ПО АСУ ТП ранее не применялись.

Основные результаты диссертационной работы используются в работе испытательной лаборатории средств защиты информации ЗАО ГИЦ ПС ВТ (Тверь).

## **Апробация результатов работы**

Полученные результаты, характеризующиеся научной новизной, подвергались обсуждению и оценке корректности на заседаниях кафедры Информационных систем (Тверь), всероссийских научно-практических конференциях (Тверь), XII международной научно-технической конференции (Смоленск), Международном конгрессе по интеллектуальным системам и информационным технологиям (Дивноморск), Российской конференции с международным участием «Искусственный интеллект, когнитивное моделирование и интеллектуальная робототехника» (Дубна), IX-й Международной научно-практической конференции "Интегрированные модели и мягкие вычисления в искусственном интеллекте" (Коломна).

## **Публикации**

По основным результатам исследования имеется семь публикаций, две из которых – в журналах из списка, рекомендованного ВАК России (в соавторстве), одна – в журнале, индексируемом в базе Scopus (в соавторстве).

## **Личный вклад автора**

Представленные на защиту результаты диссертации получены автором самостоятельно. Результаты, опубликованные совместно с другими авторами, принадлежат соавторам в равных долях. Результаты других авторов, которые использованы при изложении, содержат ссылки на соответствующие источники.

## **Структура и объем диссертации**

Диссертация включает в себя введение, четыре главы, заключение, список литературы и три приложения. Объем диссертационной работы составляет 141 страницу машинописного текста, включая 22 рисунка и 18 таблиц. Список литературы содержит 100 наименований.

## **Основное содержание диссертации**

Во введении обосновывается актуальность темы исследования, сформулированы его цель и задачи, методы исследования, научная новизна. Приводятся перечень научных результатов и положения, выносимые на защиту.

**В первой главе** приведены особенности встраиваемого ПО АСУ ТП и соответствующих промышленных производственных экспертных систем.

В данной главе приводятся требования к сертификации средств защиты информации встраиваемого ПО АСУ ТП.

Представлено описание причин возникновения решаемой в диссертационном исследовании задачи повышения эффективности процесса логического вывода в производственных экспертных ИС при сертификации встраиваемого ПО АСУ ТП.

**Во второй главе** рассматриваются известные подходы и методы решения задачи повышения скорости логического вывода производственных экспертных ИС.

В данной главе делается акцент на том, что известные подходы, которые могли бы применяться для повышения скорости логического вывода производственных экспертных ИС не подходят для использования при сертификации экспертных ИС встраиваемого ПО АСУ ТП. В связи с этим предлагается разработать собственные алгоритмы на базе АО-подхода.

Представлены разработанные алгоритмы решения задачи повышения скорости вычислительных операций в производственных экспертных ИС с использованием АО-подхода:

- одноитерационный алгоритм извлечения аспектов из последовательности фактов (ОАИ-алгоритм);

- рекурсивный алгоритм извлечения аспектов из последовательности фактов (РАИ-алгоритм).

Отмечается, что использование АО-подхода для повышения эффективности процесса логического вывода в производственных экспертных ИС встраиваемого ПО АСУ ТП ранее не выполнялось.

**Третья глава** посвящена анализу критериев повышения скорости вычислительных операций в производственных экспертных ИС встраиваемого ПО АСУ ТП.

В данной главе приведены критерии повышения скорости вычислительных операций и подробное описание использования МАИ для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества производственных правил в БЗ экспертной ИС встраиваемого ПО АСУ ТП.

Приведены алгоритм и описание каждого из этапов принятия решения согласно МАИ:

- структуризация задачи в виде иерархической структуры с несколькими уровнями;
- выполнение попарных сравнений критериев;
- вычисление коэффициентов важности для критериев;
- выполнение попарных сравнений альтернатив;
- вычисление коэффициентов важности для альтернатив;
- определение наилучшей альтернативы.

Получены значения глобальных приоритетов для БЗ экспертной ИС встраиваемого ПО АСУ ТП из 25, 50, 100, 250, 500 и 1000 производственных правил и представлены рекомендуемые для использования в каждом из случаев алгоритмы решения задачи повышения скорости вычислительных операций.

**В четвертой главе** представлено описание прототипа подсистемы интерпретации производственных правил экспертной ИС, применяемого при

сертификации встраиваемого ПО АСУ ТП, базирующегося на применении АО-подхода.

В данной главе проанализирована и описана важность сохранения исходной структуры БЗ при сертификации встраиваемого ПО АСУ ТП.

Представлены архитектура разработанного прототипа и используемые при разработке программные средства.

**В заключении** представлены основные выводы и результаты диссертационной работы. Рассматривается вопрос перспектив дальнейшего практического применения основных результатов исследования и усовершенствования разработанного прототипа подсистемы интерпретации продукционных правил экспертной ИС.

В приложения вынесены: описание тестовых наборов продукционных правил (Приложение А); блок-схема взаимодействия классов реализации подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов, построенная с помощью встроенных средств среды разработки IntelliJ IDEA (Приложение Б); исходный код подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов на языке Java (Приложение В).

# Глава 1. Анализ процесса сертификации экспертных ИС встраиваемого ПО АСУ ТП

## 1.1. Особенности встраиваемого ПО АСУ ТП

АСУ ТП, как правило, имеет многоуровневую структуру [63]. Структура типовой АСУ ТП представлена на рисунке 1.

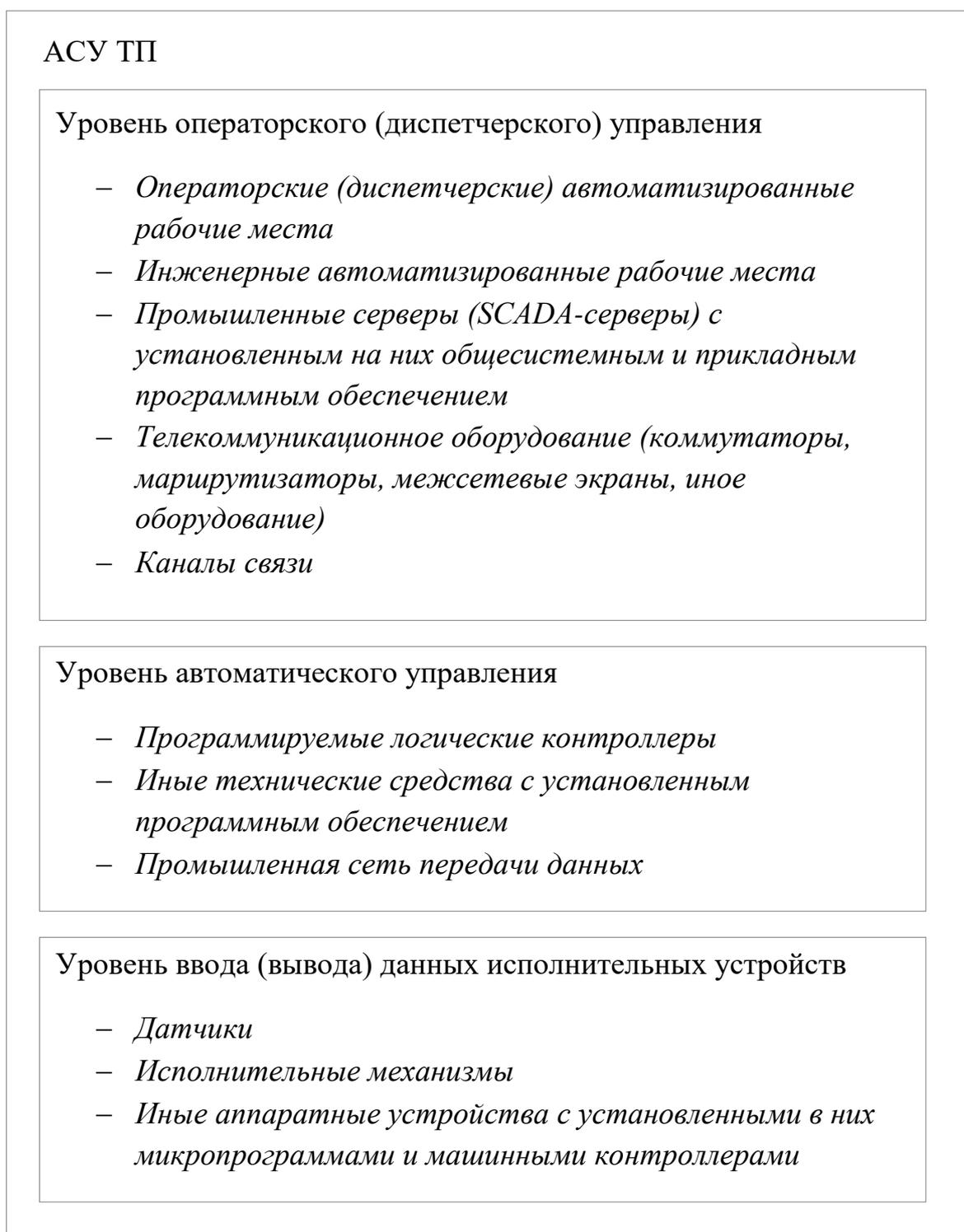


Рисунок 1. Структура АСУ ТП

Встраиваемое ПО используется на уровне автоматического управления АСУ ТП.

Встраиваемое ПО – совокупность программ на носителях данных и программных документов, предназначенная для отладки, функционирования и проверки работоспособности АС [49].

Современные ИС представляют собой сложные системы со всеми характерными особенностями сложных систем:

- наличие общей задачи;
- наличие единой цели функционирования;
- наличие иерархической системы связей;
- наличие сложности поведения системы.

Данные особенности обуславливают проблемы проектирования сложных систем ([28], [46]).

Авторы и исследователи Ю.А. Кудеяров, А.В. Шестаков, А.А. Дудыкин, А.Н. Паньков и др. выделяют следующие характерные особенности встраиваемого ПО АСУ ТП:

- значительный объем обрабатываемых информационных потоков, сложность и многогранность процессов обработки и передачи информации;
- многообразие функциональных возможностей;
- критическая важность бесперебойного поддержания работоспособности;
- болезненная, с точки зрения совместимости и технической поддержки, реакция на любые изменения в аппаратную или программную составляющие;
- необходимость особенного внимания к обеспечению триады свойств безопасности информационных ресурсов – конфиденциальности, целостности и доступности.

Встраиваемое ПО АСУ ТП, помимо выполнения своих целевых функций, должно успешно противостоять следующим угрозам обрабатываемой информации:

- неправомерный доступ
- уничтожение
- модифицирование
- блокирование
- копирование
- предоставление
- распространение
- иные неправомерные действия, в том числе деструктивные информационные воздействия (компьютерные атаки), следствием которых может стать нарушение функционирования АСУ ТП [63].

Особенностью разработки встраиваемого ПО АСУ ТП является значительная функциональная нагрузка, поэтому в соответствии с системным подходом [61], необходимо декомпозиция структуры ИС на составляющие с выделением подсистем. Свойства системы в целом при этом определяются свойствами подсистем, из которых она состоит и связей между ними [45] (системным взаимодействием всех структурных элементов [14]).

Системный подход к защите информации предполагает необходимость учёта всех взаимосвязанных, взаимодействующих и изменяющихся во времени элементов, условий и факторов, существенных для обеспечения безопасности при обработке информации с помощью встраиваемого ПО экспертных ИС [33].

Аккумуляция эмпирических знаний о принимаемых при управлении технологическими процессами решениях привело к использованию в составе встраиваемого ПО АСУ ТП производственных экспертных ИС промышленного назначения [96]. Использование экспертных ИС для сложных процессов, для которых не удаётся разработать достаточно полный алгоритм управления, позволяет облегчить труд операторов АСУ ТП ([44], [49]). Развитие

промышленных предприятий в настоящее время невозможно без применения ИС промышленного назначения [24].

Проектные пределы значений параметров выполнения целевых функций, экономические и технологические условия производства непрерывно изменяются [9], поэтому нельзя раз и навсегда определить структуру и набор знаний БЗ экспертной ИС встраиваемого ПО АСУ ТП. В следующем подразделе будут рассмотрены другие особенности промышленных производственных экспертных ИС.

## **1.2. Промышленные производственные экспертные ИС**

Промышленные экспертные ИС (expert systems, knowledge-based systems [11]) используются в условиях неопределенности состава технологических операций.

Данные ИС являются типичными системами искусственного интеллекта, в которых БЗ содержит сведения, полученные от экспертов в технологической области знаний на основе их личного опыта и неформальных рекомендаций ([42], [64]).

Другими характерными для интеллектуальных ИС признаками являются [37]:

- умение решать плохо формализуемые задачи;
- самообучение;
- адаптивность.

В целом же, задачей науки искусственный интеллект является воссоздание с помощью искусственных устройств разумных рассуждений и действий [51]. Для проблем искусственного интеллекта существенную роль играет вопрос о формальном описании тех или иных неформально поставленных задач [62].

В ходе практической деятельности человек приобретает опыт, который далее преобразуется в знания для БЗ (теоретические знания → эмпирические знания) [75]. Таким образом, поиск аналогии решения интеллектуальных задач экспертами нашёл своё отражение в экспертных ИС [22]. Стратегии экспертных ИС также основаны на знаниях человека-эксперта [52].

Решением экспертной задачи является сценарий. Сценарий должен отвечать на вопрос «что было бы, если...?», «что будет, если...?» или «что происходит в настоящий момент, если...?» [35]. При этом, добавляемые в БЗ данные должны быть формализованы и однозначно представлены [36].

Результат интереса к экспертным ИС в промышленности также продиктован их особым умением решать задачи интерпретации данных, диагностики, мониторинга, проектирования, прогнозирования и обучения [25], а также в связи с быстрым развитием компьютерного оборудования, высокоскоростных сетей, веб-программирования, распределенных и параллельных вычислений и появлением других технологий [100]. Промышленные экспертные ИС предназначены для использования их в качестве интеллектуальных ИС для реализации функций контроля и принятия решений, а также для анализа данных, поступающих от большого количества источников информации [44].

Основными особенностями экспертных ИС являются следующие [92]:

- цепочка логического вывода экспертных ИС может быть составлена из небольших промежуточных подцелей, которые легче доказать;
- способность экспертных ИС рассуждать с помощью правил и данных, которые точно не известны;
- наличие методов объяснения – способность экспертных ИС представить процесс рассуждения, который она использовала при логическом выводе.

Экспертные ИС обрабатывают информацию в символической форме и успешно справляются с ошибками в данных и с несовершенными правилами рассуждения, могут одновременно выдерживать множество конфликтующих гипотез [96].

Наиболее известные экспертные ИС [77]:

- CLIPS;
- OpenCyc;
- WolframAlpha;
- MYCIN;

- HASP/SIAP;
- Акинатор;
- IBM Watson;
- OPS 5;
- HEARSAY-II [68];
- R1/XCON [30];
- Exsys Corvid [80].

Основные свойства экспертных ИС (Рисунок 2) [58]:

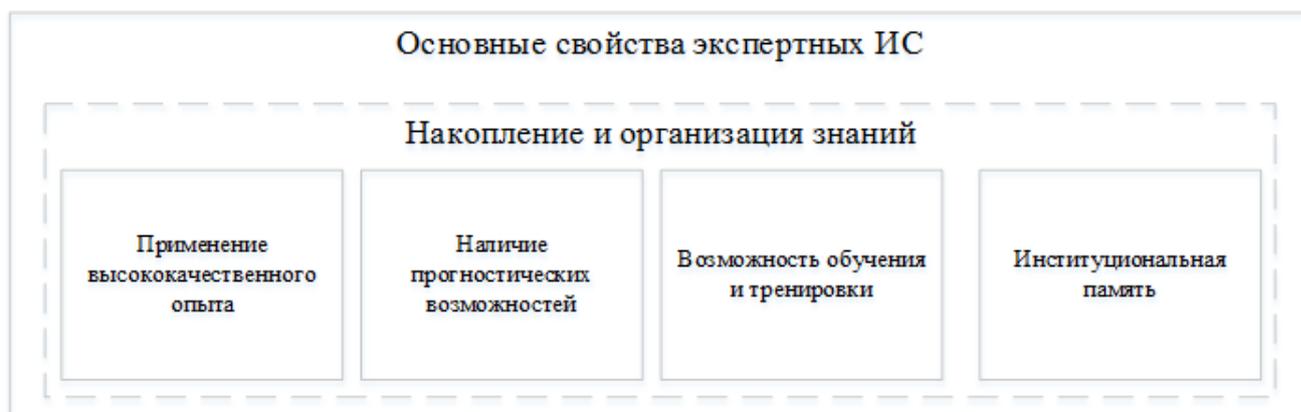


Рисунок 2. Основные свойства экспертных ИС

Каждая экспертная ИС базируется на модели представления знаний. К наиболее общим моделям относятся:

- логическая;
- продукционная;
- фреймовая;
- семантическая сеть;
- комбинированная.

Наиболее распространено применение продукционной модели [18].

Понятие продукционной модели ввели в начале 1970 года А. Ньюэлл [Allen Newell] и Х.А. Саймон [Herbert A. Simon] изучая психологическую модель поведения человека [90].

В этой модели порознь взятые части человеческого знания были представлены в отдельных единицах, называемых правилами продукций или продукционными правилами (production rule).

Впервые системы, основанные на продукционных правилах, были предложены в 1943 году Эмилем Постом (Emil Leon Post).

Продукционные системы представляют знания о решении задач в виде правил продукции – выражений следующего вида, например,

«ЕСЛИ [условие], ТО [действие]» ([3], [89]).

К продукционным правилам применимы логические операции: И, ИЛИ, НЕ. Например,

«ЕСЛИ [условие1] И [условие2] И [условие3] ... ТО [действие1], [действие2], ...» [96].

БЗ продукционных экспертных ИС состоит из фактов (данных) и правил (эвристик), которые представляют факты в порядке, необходимом для решения поставленной задачи (проблемы) ([23], [53]).

В продукционном правиле антецедентом (Antecedent) называется условная часть, консеквентом (Consequent) называется часть действия. Для сопоставления антецедентов и выполнения консеквентов предназначен интерпретатор правил.

Правила продукции наиболее естественная форма представления знаний, обладающая следующими положительными свойствами [7]:

- модульность (относительная независимость отдельных фрагментов знаний);
- наращиваемость (ввод в БЗ новых правил);
- модифицируемость (независимая, в силу модульности, модификация правил);
- прозрачность (наличие механизма разъяснения принятых решений).

База данных (БД) продукционной системы в наиболее абстрактном виде (Рисунок 3) состоит из набора правил (таблица RULE), набора исходных фактов (таблица FACT), набора действий (таблица ACTION), фактов, которые составляют базу антецедентов (таблица ANTECEDENT) и действий, образующих базу консеквентов (таблица CONSEQUENT).

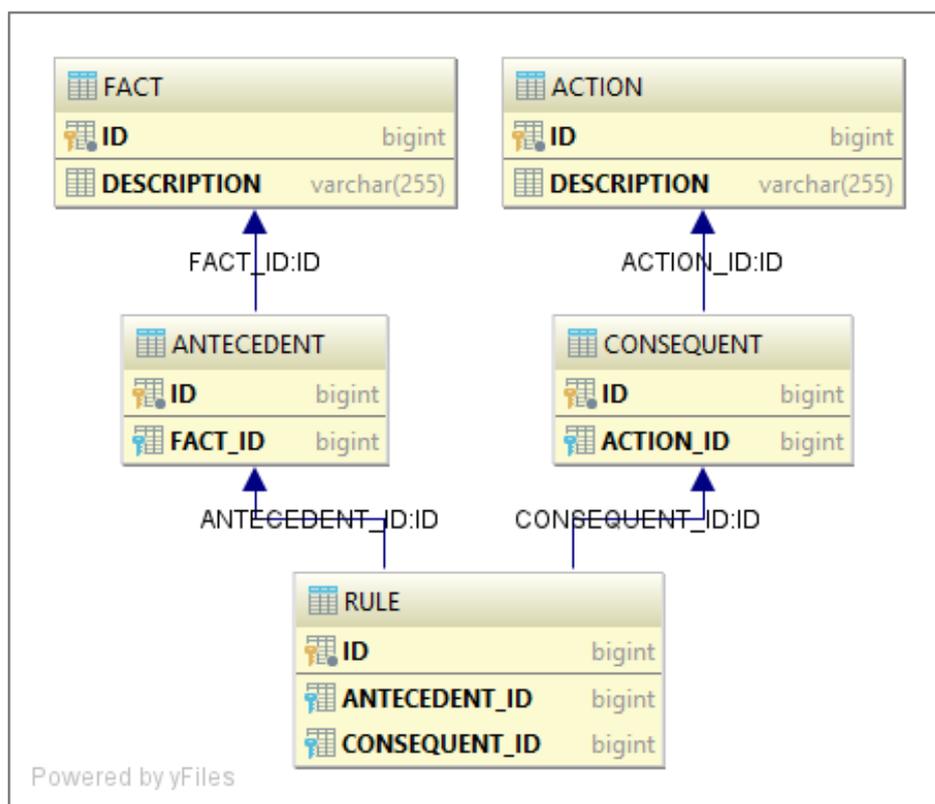


Рисунок 3. Общая схема БД продукционной экспертной ИС

Для программирования экспертных ИС используются языки программирования. Список используемых языков программирования приведён в таблице 1.

Таблица 1. Языки программирования экспертных ИС

№ п/п	Тип ЯВУ	Примеры	Применимость	Недостатки
1	Языки разработки порождающих правил	CLIPS	Хорошо согласуется с представлением знаний в форме логического вывода (Если-то)	Сложность организации эффективного процесса обработки правил
2	Объектно-ориентированные языки	LOOPS	Подходит для описания сложных объектов и отношений между ними	Сложность представления программного объекта по отношению к предметной области

№ п/п	Тип ЯВУ	Примеры	Применимость	Недостатки
3	Языки логического программирования	PROLOG, LISP	Эмулирование простого логического вывода	Сложность создания механизма обработки противоречий

Преодоление недостатков отдельных языков возможно за счёт объединения их с другими парадигмами. Например, добавление объектно-ориентированного расширения COOL в средство CLIPS [69].

В АСУ ТП могут применяться следующие классификации экспертных ИС по связи с реальным временем [44]:

- статические экспертные ИС, которые разрабатываются для предметных областей, в которых БЗ и интерпретируемые данные не меняются во времени (стабильны);
- квазидинамические экспертные ИС, интерпретирующие ситуации, которые меняются с некоторым фиксированным значением интервала времени;
- динамические экспертные ИС, образующие связь в режиме реального времени с датчиками объектов, выполняющие непрерывную интерпретацию входной информации.

Типовая структура экспертной ИС встраиваемого ПО АСУ ТП представлена на рисунке 4.

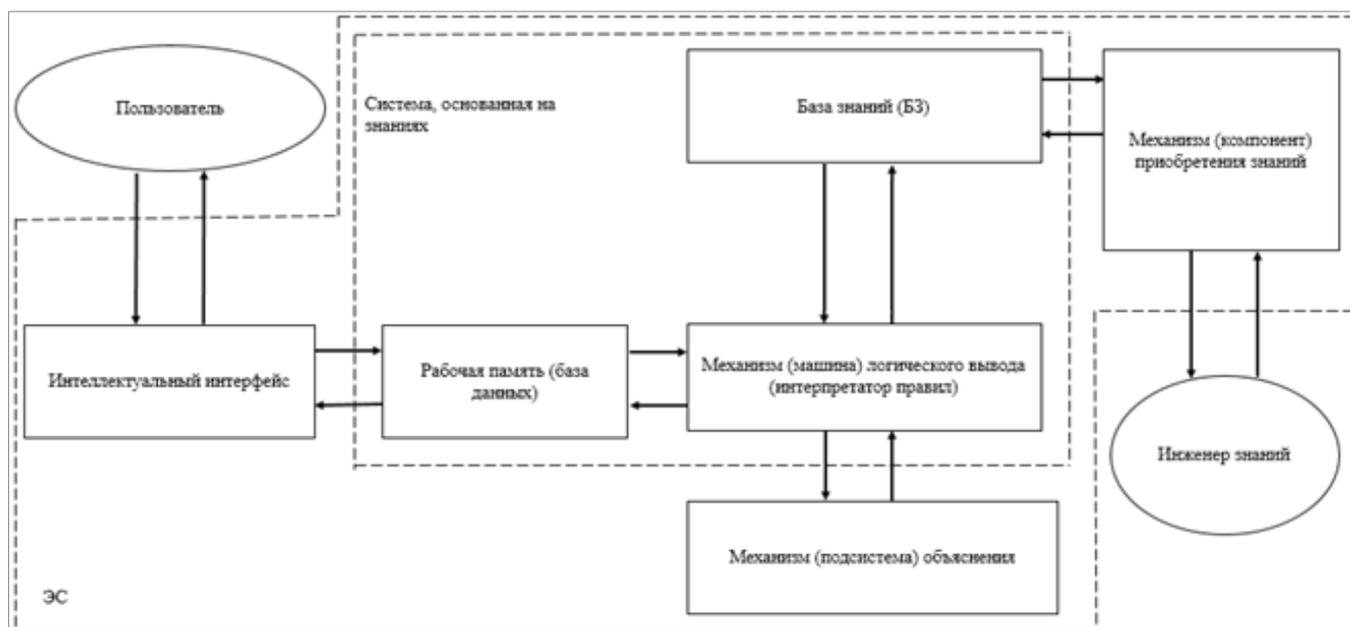


Рисунок 4. Типовая структура экспертной ИС встраиваемого ПО АСУ ТП

Эксперт независимо от внешних влияний и собственной выгоды высказывает суждения из области его специальных знаний и/или практического опыта по поставленным перед ним вопросам [72].

Одними из главных преимуществ экспертных ИС перед человеком-экспертом являются постоянство, лёгкость передачи информации, устойчивость перед влиянием внешних факторов и их стоимость. Более подробное сравнение экспертных ИС и людей-экспертов представлено в таблице 2.

Таблица 2. Сравнение экспертных ИС и людей-экспертов

№ п/п	Компетентность	
	Искусственная	Человеческая
1	Постоянная	Непрочная
2	Легко передаваемая	Трудно передаваемая
3	Легко документированная	Трудно документированная
4	Устойчивая	Непредсказуемая
5	Приемлемая по затратам	Дорогая
6	Запрограммированная	Творческая
7	Нуждается в подсказке	Приспосабливающаяся
8	Использует символьный ввод	Использует чувственное восприятие
9	Узконаправленная	Широкая по охвату

№ п/п	Компетентность	
	Искусственная	Человеческая
10	Использует специализированные знания	Использует общедоступные знания

Разработка экспертных ИС – это активно развивающаяся ветвь информатики, направленная на использование ЭВМ (электронно-вычислительных машин) для обработки информации в тех областях науки и техники, где традиционные математические методы моделирования малоприменимы, где важны смысловая и логическая обработка информации, опыт экспертов ([12], [79]).

Согласно [63], для экспертных ИС встраиваемого ПО АСУ ТП возникает необходимость осуществления контроля, которая находит своё отражение в процедуре сертификации – формы подтверждения соответствия продукции требованиям технических регламентов [74].

В следующем подразделе будут приведены требования к сертификации производственных экспертных ИС встраиваемого ПО АСУ ТП.

### **1.3. Формирование требований к сертификации промышленных производственных экспертных ИС**

Сертификация средств защиты информации производится в соответствии с утвержденным приказом № 55 ФСТЭК России от 3 апреля 2018 г. «Положением о системе сертификации средств защиты информации» [66].

Для обеспечения защиты информации в АСУ ТП должны проводиться следующие мероприятия [63]:

- формирование требований к защите информации в АСУ ТП;
- разработка системы защиты АСУ ТП;
- внедрение системы защиты АСУ ТП и ввод данной системы защиты в действие;
- обеспечение защиты информации при эксплуатации АСУ ТП;
- обеспечение защиты информации после эксплуатации АСУ ТП.

Как видно из представленного, экспертная ИС, даже являясь частью встраиваемого ПО АСУ ТП, должна учитывать требования безопасности информации к защищаемой информации [27].

Основная цель внедрения АСУ ТП – это повышение экономической эффективности предприятия, поэтому при сертификации встраиваемого ПО АСУ ТП на первый план выступают сроки проведения всех необходимых проверок. Согласно [66], сертификационные испытания проводятся в сроки, установленные договором, заключенным заявителем с испытательной лабораторией.

Требование важности сосредоточения на уменьшении продолжительности сертификационных испытаний обусловлено невозможностью снижения конкурентоспособности предприятия. Кроме того, уменьшение сроков сертификационных испытаний ведёт, соответственно, к уменьшению стоимости проведения сертификационных испытаний.

Другим требованием (особенностью), которое логично предъявить к сертификационным испытаниям, является необходимость в неизменности параметров технологического процесса, показателей надежности [32] и производительности [41], а также пределов значений параметров выполнения целевых функций, заложенных в БЗ производственной экспертной ИС встраиваемого ПО АСУ ТП.

Принятие (формирование) решений экспертными ИС промышленного назначения происходит в результате логического вывода путём использования БЗ. В следующем подразделе будут описаны особенности логического вывода производственных экспертных ИС промышленного назначения.

#### **1.4. Особенности логического вывода производственных экспертных ИС промышленного назначения. Постановка задачи исследования**

Промышленность России представлена множеством отраслей и предприятий. Сложившееся обвальное нарастание массы самой разнообразной информации о технологиях промышленности, знаний и метазнаний о промышленных процессах

наделяет современные продукционные экспертные ИС встраиваемого ПО АСУ ТП большим количеством правил (продукций).

Многие авторы (Валетов В.А., Орлова А.А., Третьяков С.Д. и др.) рассматривали проблемы экспертных ИС и в качестве одного из актуальных недостатков продукционных экспертных ИС выделяют меньшую эффективность процесса логического вывода, по сравнению с другими моделями представления знаний.

Причину этого можно найти в том, что при логическом выводе требуется значительное количество времени на перебор и непроизводительные проверки применимости правил [8]. Кроме того, при большом объёме знаний усложняется понимание взаимосвязей объектов [78].

Из [51] в качестве недостатка промышленных экспертных систем можно привести следующее утверждение: увеличение времени вычислений и получения выводов происходит вместе со значением произведения числа правил на число записей в базе фактов.

В [16] отмечается недостаток переполненной продукциями продукционных экспертных ИС, заключающийся в присущей недетерминированности (неоднозначности), из-за чего возникают принципиальные трудности при проверке корректности работы системы. При большом количестве правил становится сложно отслеживать непротиворечивость базы знаний [60].

Кроме того, ранее представленные требования к встраиваемому ПО АСУ ТП показывают важность и актуальность проблемы повышения эффективности процесса логического вывода в продукционных экспертных ИС.

Таким образом, решение задачи повышения эффективности процесса логического вывода в продукционных экспертных ИС при сертификации встраиваемого ПО АСУ ТП является актуальным.

### **1.5. Выводы по главе**

В данной главе были описаны особенности встраиваемого ПО АСУ ТП и промышленных продукционных экспертных ИС.

Были сформулированы следующие требования (особенности) предъявляемые к сертификации производственных экспертных ИС встраиваемого ПО АСУ ТП:

- уменьшение продолжительности сертификационных испытаний;
- необходимость в неизменности параметров технологического процесса и пределов значений параметров выполнения целевых функций, заложенных в БЗ производственной экспертной ИС встраиваемого ПО АСУ ТП.

Были описаны особенности логического вывода производственных экспертных ИС промышленного назначения и их главные недостатки, среди которых:

- значительное количество времени на перебор и непроизводительные проверки применимости правил (продукций);
- сложность понимания взаимосвязей объектов;
- продолжительность вычислений и времени логического вывода увеличивается вместе с количеством числа правил и число записей, находящихся в базе фактов;
- недетерминированность (неоднозначность);
- при большом количестве правил становится сложно отслеживать непротиворечивость базы знаний.

Исходя из этого, была поставлена задача исследования, заключающаяся в нахождении методов повышения эффективности процесса логического вывода в производственных экспертных ИС при сертификации встраиваемого ПО АСУ ТП.

## **Глава 2. Обоснование методов решения задачи повышения скорости логического вывода продукционных экспертных ИС промышленного назначения**

### **2.1. Известные подходы повышения скорости логического вывода промышленных продукционных экспертных ИС**

#### **2.1.1. Фильтры**

В работе [51] для уменьшения количества безуспешных попыток, предпринимаемых для выполнения продукционных правил в экспертной ИС, предлагается использовать разнообразные фильтры, которые перед унификацией будут исключать из БЗ бесполезные (ненужные в данный момент) правила и факты. В работе [91] предлагаются четыре подобных фильтра и приводятся соответствующие формулы оценки времени их работы.

#### **2.1.2. Фреймы**

В работе [93] для уменьшения числа попыток проверки применимости правил предлагается автоматическая организация элементарных фактов в более сложные структуры – фреймы.

Фреймы позволяют получить представление более высокого уровня. Однако, при этом вместе с увеличением разнообразия рассматриваемых объектов происходит уменьшение вероятности успеха унификации и одновременно уменьшение числа бесполезных попыток проверки применимости правил [51].

#### **2.1.3. Дуальные экспертные системы**

Согласно [34], исходную непроизводительную экспертную ИС можно преобразовать в дуальную экспертную ИС, в которой содержатся и совместно используются две машины вывода: дедуктивная, использующая продукции, содержащиеся в БЗ, и индуктивная, обеспечивающая обобщение на основе накопленных в БД прецедентов. Совместное использование двух методов принятия решений расширяет возможности построения правил вывода за счет использования дополнительной информации и повышает надежность результатов.

#### **2.1.4. Нейронные сети**

Экспертные ИС, в некоторой степени, воспроизводят осознанные мыслительные процессы человека, следовательно, к ним применимы традиционные нейросетевые модели (перцептрон Розенблатта [26], нейронная сеть Хопфилда [3] и др.).

Особенностью любой нейронной сети, является их способность к самостоятельному обучению, обладание возможностью действовать на основании предыдущего опыта, делая каждый раз всё меньше ошибок [26], возможность воспринимать новые знания без потери прежних (дилемма стабильности-пластичности) [1].

К реальным структурам нервной системы нейронные сети имеют лишь косвенное отношение ([5], [40]).

Нейронные сети могут использоваться для извлечения правил из обученных нейросетей для использования в экспертных ИС или включения знаний о предметной области экспертной ИС в нейросеть для их совершенствования ([4], [54]).

#### **2.1.5. Генетические алгоритмы**

Согласно [65], генетические алгоритмы – это адаптивные методы поиска, которые используются для решения задач оптимизации. Генетическое программирование – одно из направлений эволюционного поиска и ориентировано в основном на решение задач автоматического синтеза программ [15].

Генетические алгоритмы могут использоваться для повышения эффективности обучения нейронной сети, решающей задачу формирования БЗ экспертной ИС [43]. Оптимальное решение ищется в серии последовательных приближений аргументов с лучшими значениями функций приспособленности, генерируемых в результате мутации и скрещивания хромосом [70].

### **2.1.6. Rete-алгоритм**

Разработанный Чарльзом Форги [Charles Forgy], алгоритм Rete (Rete-алгоритм) — представляет собой эффективный алгоритм для сопоставления с образцом, используемый в продукционных экспертных ИС [82].

Как упоминалось выше, в большом количестве современных экспертных ИС логическая пригодность правил продукции рекурсивно проверяется для каждого правила продукции к каждому факту БЗ, что, очевидно, выполняется недопустимо медленно для встраиваемого ПО АСУ ТП.

Rete-алгоритм позволяет повысить эффективность логического вывода.

При внедрении Rete-алгоритма экспертная ИС формирует префиксное дерево (или граф), узлам которого соответствуют антецеденты. Таким образом корневой путь к листу представляет полное условие логического вывода отдельного продукционного правила.

В процессе работы экспертной ИС с применением Rete-алгоритма каждый отдельный узел представлен набором отдельных фактов, входящих в условие. При редактировании набора фактов отмечается каждый соответствующий факту узел во всём префиксом дерева (или графе). Соответственно, правило продукции будет выполнено при прохождении экспертной ИС корневой пути к листу префиксного дерева (или графа).

Для современных АСУ ТП, обладающих большим количеством продукционных правил, алгоритм Rete-алгоритм используется редко из-за большой потребности в оперативной памяти.

### **2.2. Подтверждение неприменимости известных подходов повышения скорости логического вывода промышленных продукционных экспертных ИС при сертификации встраиваемого ПО АСУ ТП**

Среди предъявленных требований к сертификации экспертных ИС встраиваемого ПО АСУ ТП была отмечена важная необходимость в неизменности параметров технологического процесса и пределов значений параметров выполнения целевых функций, заложенных в БЗ продукционной экспертной ИС встраиваемого ПО АСУ ТП.

В соответствии с этим требованием, при сертификации *не могут быть применены* такие известные подходы повышения скорости логического вывода промышленных производственных экспертных ИС как:

- фильтры;
- фреймы;
- дуальные экспертные системы;
- нейронные сети;
- генетические алгоритмы.

Rete-алгоритм *не применим* в силу того, что используется редко из-за большой потребности в оперативной памяти.

В следующем подразделе будет рассмотрен аспектно-ориентированный подход, который как раз и предназначен для применения в целях неизменности функциональности, что удовлетворяет главному требованию, предъявляемому к сертификации экспертных систем встраиваемого ПО АСУ ТП.

### **2.3. Алгоритмы решения задачи повышения скорости вычислительных операций в производственных экспертных ИС с использованием аспектно-ориентированного подхода**

#### **2.3.1. Особенности аспектно-ориентированного подхода**

В конце 1990-х гг. была предложена новая методология – аспектно-ориентированный подход к разработке программного обеспечения (АОП).

В основе аспектно-ориентированного программирования лежит понятие cross-cutting concerns, которое не имеет устоявшегося эквивалента в русском языке. Наиболее близким по смыслу считается и чаще всего используется словосочетание «сквозная функциональность». Под сквозной функциональностью понимается функциональность, реализовать которую в отдельном компоненте языка программирования традиционными средствами процедурного или объектно-ориентированного программирования или очень сложно, или невозможно, поскольку эта функциональность необходима в большей части модулей системы. Кроме того, эта функциональность не

относится напрямую к предметной области. Примером такой функциональности является протоколирование работы системы (logging).

Использование АОП помогает следовать принципу разделения ответственности (separation of concerns), что положительно сказывается на многих характеристиках приложений, некоторыми из них являются:

- более качественная реализация;
- улучшенные возможности тестирования;
- модульность системы.

АОП предназначен для модуляризации сквозной функциональности (cross-cutting concerns) и её автоматизированного, безопасного и надёжного внедрения в необходимую программу. Кроме того, АОП позволяет выполнять поиск и видоизменять в программе уже реализованную сквозную функциональность – функциональность, рассредоточенную по всему коду программы.

Целью АОП является явное разделение и комплексирование аспектов – логически замкнутых и технологически однородных единиц функциональности, реализующих конкретные классы задач. В качестве аспектов выступают средства реализации как функциональных возможностей (например, сбор данных, их хранение, вычислительная обработка и т.д.), так и нефункциональных требований (например, координация функционирования распределенных компонентов, защита информации и т.д.). Разделение аспектов и представление их в виде отдельных единиц комплексирования, связываемых в явно заданных точках, значительно упрощает компоновку и сопровождение системы. Аспектная декомпозиция облегчает организацию деятельности коллектива разработчиков, являющихся узкими специалистами.

Используя АОП, аспекты проектируются и реализуются независимо друг от друга, объединяясь в одно целое, согласно свойству системности, в автоматическом режиме [2].

Методология АОП была предложена в 1996 году группой инженеров исследовательского центра Xerox PARC в Исследовательском центре Пало-Альто

под руководством Грегора Кичалеса (Gregor Kiczales) в работе «Aspect-oriented programming» ([31], [94]).

По мнению автора АОП, основным принципом появления ранее неизвестных теорий программирования можно назвать принцип расщепления программы на отдельные модули [87]. Однако, в настоящий момент, ни одна из существующих парадигм не предложила способа разделения реализации различных аспектов одного приложения.

Понятие «аспект» (лат. *aspectus* – вид) понимается так же, как в работе [84], а именно как точка зрения на изучаемый предмет, рассматриваемая в изоляции от других точек зрения. Так, один из аспектов приложения (помимо основного назначения) – это безопасность (защищённость от конкретных атак).

Если, в силу декомпозиции системы, один и тот же модуль должен реализовывать алгоритмы двух или более аспектов, то говорят о пересекающихся аспектах. АОП – это парадигма программирования, целью которой является введение уровня модульности, позволяющего реализовывать различные аспекты системы изолированно друг от друга, независимо от того, пересекаются ли они или нет при первоначальной декомпозиции [81].

Реализация ПО с использованием АОП заключается в раздельном описании каждого аспекта системы и указании способа соединения различных аспектов в единое приложение [83]. Соединение различных аспектов в приложение осуществляется автоматически, средствами аспектно-ориентированных средств разработки.

АОП не является заменой какой-либо другой парадигмы программирования, но её расширением [84]. Типичный процесс создания приложения с помощью аспектно-ориентированных технологий выглядит следующим образом: производится декомпозиция системы на модули привычными (не аспектно-ориентированными) методами, выделяются модули, по которым пересекаются несколько аспектов приложения, один аспект объявляется главным (он выполняет основное назначение системы; такой аспект в будущем будем называть главным) –

он реализуется привычными средствами (без АОП), прочие аспекты реализуются с помощью АОП.

АОП – одна из главных современных парадигм программирования. Она вводит уровень модульности, позволяющий изолированно разрабатывать различные аспекты приложения [83].

Как каждый технологический процесс можно дифференцировать рядом более простых, но связанных составляющих [48], так и функциональность всей ИС можно дифференцировать на модули.

АОП представляя собой современное развитие объектно-ориентированного подхода к программированию, предназначено для извлечения сквозных функций внутри бизнес-логики ИС. Бизнес-логика представляет собой описание набора бизнес-правил, использующихся в ходе производственной деятельности промышленных предприятий [17].

Сквозная функциональность (Cross-cutting concern) – это функциональность, рассеянная по всему исходному коду ПО, систематически независимая от предметной области. АОП предоставляет возможность вызывать код сквозных функций без изменения исходного кода программы в определенный момент работы программы [17]:

- перед вызовом метода;
- после вызова метода, независимо от результата;
- после успешного вызова (выполнение метода завершилось успешно);
- после исключения (вызванный метод возбудил исключение);
- вокруг (до и после вызова метода).

К общесистемным требованиям, обычно относятся, перечисленные на рисунке 5, следующие сквозные функции [17]:



Рисунок 5. Сквозная функциональность в промышленных ИС

К сокращению количества вариантов сквозной функциональности приводит реализация главной цели АОП, которая состоит в том, что типовой код (не связанный с решением прикладных задач предметной области и повторяющийся от компонента к компоненту) не должен смешиваться с прикладной бизнес-логикой [17].

В таблице 3 сформулированы основные недостатки и преимущества АОП с точки зрения практической ценности промышленного предприятия [17].

Таблица 3. Недостатки и преимущества АОП

№ п/п	Недостатки	Преимущества
1	Отсутствие структурированной информации, о разработке ПО с использованием данной методологии	Более прозрачный код логики (уменьшение связанности классов)

№ п/п	Недостатки	Преимущества
2	Высокий порог вхождения (трудности с пониманием концепции)	Легкость модификации кода
3	Ограниченная помощь среды разработки (IDE)	Меньшая вероятность возникновения ошибок
4	Отсутствие подробной документации на русском языке	Соблюдение принципа DRY (англ. «Don't repeat you're self» [98] – не повторяйся)
5	Незначительное снижение скорости выполнения кода	Уменьшение шаблонных ошибок
6		Упрощение разработки документации, и, как следствие, улучшение ее читаемости
7		Улучшение модульности ИС (модули, содержат только базовую функциональность, а вторичные проблемы вынесены в аспекты)
8		Упрощение модульного тестирования (unit testing)

Таким образом, можно констатировать тот факт, что главным преимуществом и пользой АОП является повышение прозрачности кода бизнес-логики и значительное упрощение прикладных модулей в силу того, что они будут представлены только базовой функциональностью, а вторичные проблемы будут абстрагированы в отдельные аспекты.

На рисунке 6 представлены пространство требований и реализации:

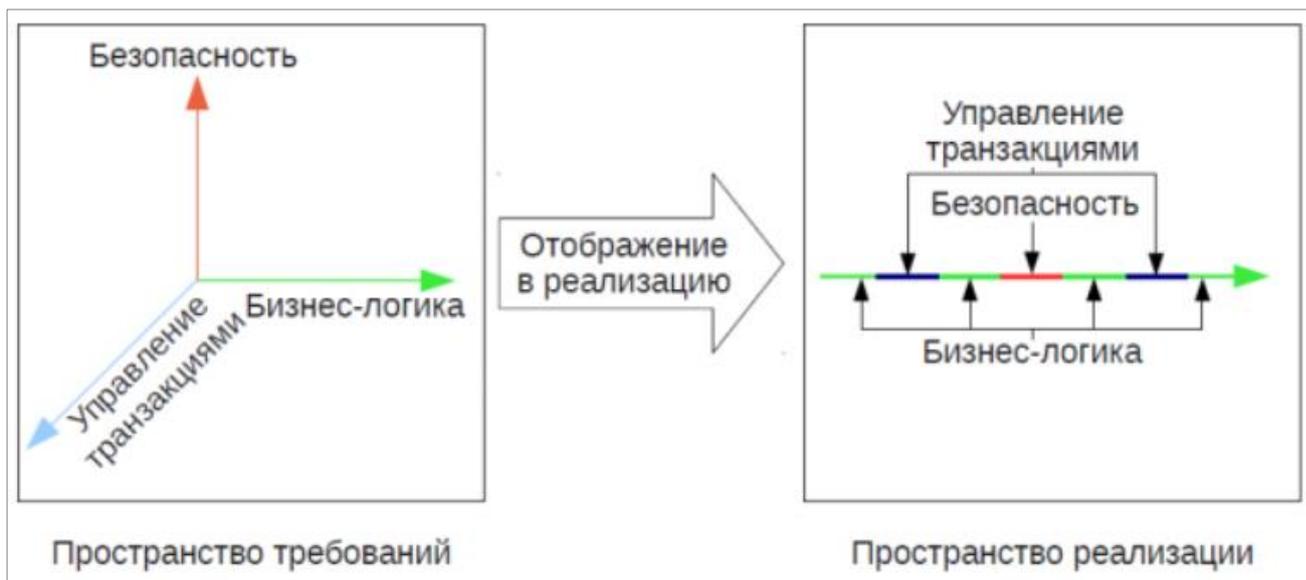


Рисунок 6. Сворачивание многомерного пространства требований в одномерное пространство реализации

Пример сквозной функциональности в АСУ ТП представлен на рисунке 7.

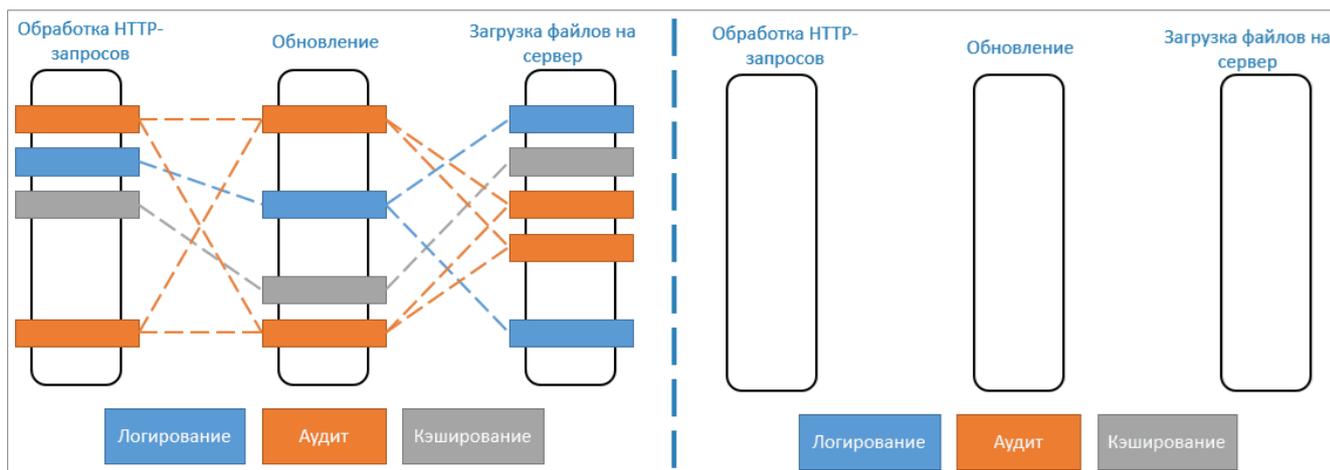


Рисунок 7. Пример сквозной функциональности в АСУ ТП

### ***Фундаментальные понятия АОП [88]***

*JoinPoint* (точка соединения, точка прикрепления, точка выполнения) – ассоциация контекста выполнения (например, вызова метода) в строго определённой единственной точке выполнения программы.

*Pointcut* (срез) – набор точек выполнения *JoinPoint*, удовлетворяющих заданному условию.

*Advice* (рекомендация, совет, инструкция) – набор инструкций, который выполняется (до, вместо или после) точки выполнения, составляющей заданный срез.

- *before* – выполняется перед выполнением инструкций, входящих в данную точку выполнения;
- *after returning* – выполняется после успешного возвращения значения из описываемой точки выполнения;
- *after throwing* – выполняется после возникновения исключительной ситуации в описываемой точке выполнения;
- *after* – выполняется после возвращения из описываемой точки выполнения в любом случае;
- *around* – выполняется вместо описываемой точки выполнения.

*Aspect* (аспект) – главная единица модульности АОП [85], инкапсулирующая срезы точек выполнения (Pointcut) и наборы инструкций (Advice).

*Introduction* (внедрение) - изменение структуры элемента программы т.е. добавление к нему поддержки интерфейсов, новых методов, свойств, полей, параметров и т.п.

*Weaving* (связывание) – процесс интеграции правил вплетения кода аспектов с кодом, ответственным за бизнес-логику, результатом которого является конечная система.

Таким образом, основные концепции АО-подхода – это:

- аспект или модульный элемент (Aspect);
- совет или конструкция изменения поведения системы (Advice);
- точка соединения или точка вызова (Join point);
- срез или конструкция отбора точек соединения (Pointcut);
- цель (Target);
- компоновщик или аспектный интегратор (Weaver).

Взаимосвязь концепций АО-подхода в упрощенном виде представлена на рисунке 8.

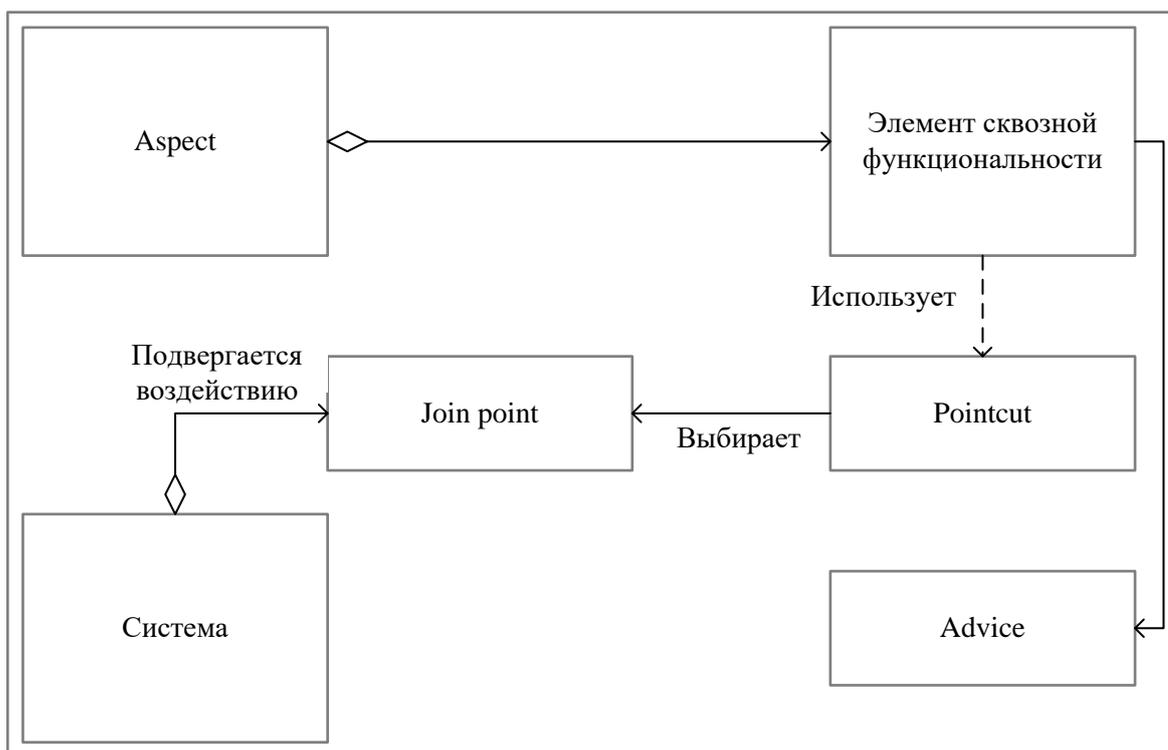


Рисунок 8. Упрощенная схема АО-подхода

Для связывания аспектов с системой предназначен компоновщик. Результатом связывания является целевой объект (цель). В общем случае целевой объект складывается из бизнес-логики системы и сквозной функциональности, представленной аспектом.

В настоящее время сформировался подход AOSD (Aspect-oriented software development) – комплексный АО-подход к сбору требований, построению архитектуры и дизайну систем. Данный подход формируют:

- аспектно-ориентированная инженерия требований;
- аспектно-ориентированное управление бизнес-процессами;
- аспектно-ориентированная системная архитектура;
- аспектно-ориентированное моделирование и дизайн;
- аспектно-ориентированное программирование (АОП).

АОП наиболее известная область применения АО-подхода. Согласно данным Google Trends (Рисунок 9) популярность данной техники в последние 5 лет неизменно находится на достаточно высоком уровне.

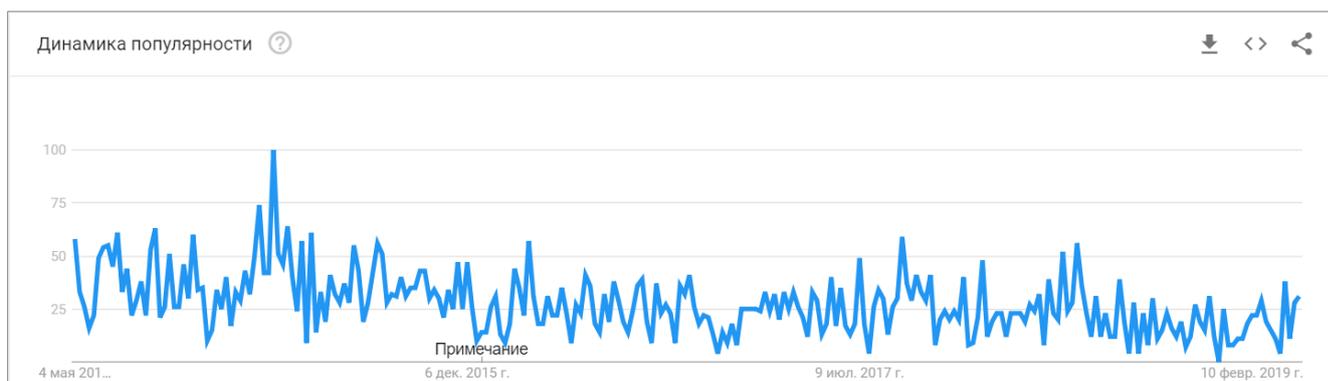


Рисунок 9. Динамика популярности АОП за последние 5 лет согласно Google Trends

Альтернативами АОП являются паттерны проектирования и DI/IoC-фреймворки.

В ходе процесса разработки ИС разработчики часто используют так называемые паттерны проектирования (англ. Design Patterns).

Паттерн проектирования – описание взаимодействия объектов и классов, адаптированных для решения задачи проектирования в конкретном контексте ([84], [87]).

Паттерны проектирования предназначены для возможности получить более компактную, простую и понятную архитектуру программы [13].

Паттерны проектирования именуется, абстрагируют и идентифицируют ключевые аспекты структуры общего решения, позволяя применить разработанную с учетом паттернов проектирования архитектуру для создания повторно используемого дизайна архитектуры программ [39]. При этом сами паттерны обладают всеми теми недостатками, что и любой другой объектно-ориентированный код: реализация «сквозной функциональности» и «размазанность» исходного кода ([84], [87]).

Инверсия управления (англ. Inversion of Control, IoC) – принцип объектно-ориентированного программирования, используемый для уменьшения сцепленности и запутанности в компьютерных программах [73]. Также архитектурное решение интеграции, упрощающее расширение возможностей

системы, при котором контроль над потоком управления программы остаётся за каркасом [86].

Инверсия управления – это некий абстрактный принцип, набор рекомендаций для написания слабо связанного кода. Суть которого в том, что каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали конкретной реализации других компонентов.

Внедрение зависимостей (англ. Dependency injection, DI) – процесс предоставления внешней зависимости программному компоненту и одна из реализаций этого принципа (помимо Factory Method, Service Locator).

IoC-контейнер – это фреймворк (библиотека), позволяющая выполнить в разумных пределах упрощение и автоматизацию написания кода через использование данного подхода.

В следующем подразделе будет представлено описание разработанных алгоритмов решения задачи повышения скорости вычислительных операций в производственных экспертных ИС встраиваемого ПО АСУ ТП, базирующихся на применении АО-подхода.

АО-подход представляет собой средство консолидации сквозных фактов в исходном наборе производственных правил [20], позволяя таким образом абстрагировать в отдельные аспекты множественные иерархии сквозных фактов и мультисвязи между производственными правилами [21].

Использование АО-подхода для повышения эффективности процесса логического вывода в производственных экспертных ИС ранее не выполнялось.

К разработке алгоритмов извлечения аспектов из последовательности фактов было предъявлено следующее требование: одновременное сокращение количества частных сведений при использовании АО-подхода не должно уменьшать информативность всего набора производственных правил [6].

### 2.3.2. Одноитерационный алгоритм извлечения аспектов из последовательности фактов

Одноитерационный алгоритм извлечения аспектов из последовательности фактов (ОАИ-алгоритм) позволяет упростить правила БЗ путём выделения аспектов.

Особенности полученных в результате применения ОАИ-алгоритма правил:

- меньшее количество сквозных фактов, по сравнению с исходным набором правил;
- средний уровень абстракции.

Описание ОАИ-алгоритма:

1. Получить исходный набор продукционных правил (базу правил):

*БЗ представлена непустым множеством правил:*

$$R = \{rule_1, rule_2, \dots, rule_n\} \text{ (1)},$$

где  $R \neq \emptyset$ ,  $n \in N$  – количество правил в БЗ.

2. Выделить из набора продукционных правил множество антецедентов:

*Поскольку каждое правило включает в себя антецедент, тогда:*

$$ANT = \{ant_1 \in rule_1, ant_2 \in rule_2, \dots, ant_n \in rule_n\} \text{ (2)}.$$

3. Дифференцировать отдельные факты из антецедентов:

*Каждый антецедент представлен набором известных фактов, соединенных с помощью логических операций, тогда:*

$$\begin{aligned} ANT &= \{ant_1 = \{fact_1, fact_2, \dots, fact_m\} \\ &\subset F_1, ant_2 = \{fact_1, fact_2, \dots, fact_m\} \\ &\subset F_2, \dots, ant_n = \{fact_1, fact_2, \dots, fact_m\} \\ &\subset F_n\} \text{ (3)}, \end{aligned}$$

где  $m \in N$  – количество фактов в БЗ,  $F_1, F_2, \dots, F_n$  – множество фактов отдельных антецедентов, причём:

$$F = F_1 \cup F_2 \cup \dots \cup F_n \text{ (4)},$$

где  $F \neq \emptyset$ ,  $F$  – множество всех фактов в БЗ.

4. Выполнить поиск аспектов среди отдельных фактов:

*Аспектом (Aspect)  $asp_i$  будем называть пересечение отдельных фактов в независимых антецедентах, т.е.:*

$$asp_i = F_1 \cap F_2 \vee F_1 \cap F_3 \vee F_1 \cap F_4 \vee F_2 \cap F_3 \vee F_2 \cap F_4 \vee \dots \vee F_{n-1} \cap F_n \quad (5),$$

где  $i \in N$  – номер выявленного аспекта.

5. Исходя из результатов поиска оформить набор аспектов:

*Множество аспектов представлено следующим образом:*

$$ASP = \{asp_1, asp_2, \dots, asp_k\} \quad (6),$$

где  $k \in N$  – количество выявленных аспектов.

6. Выполнить замену фактов в исходном наборе продукционных правил сформированными на предыдущем шаге аспектами:

*Каждое множество фактов отдельных антецедентов, в случае выявления аспекта, уменьшится на количество фактов, входящих в аспект:*

$$\begin{aligned} F_1^* &= F_1 \setminus asp_1 \vee F_1 \setminus asp_2 \vee \dots \vee F_1 \setminus asp_k, \\ F_2^* &= F_2 \setminus asp_1 \vee F_2 \setminus asp_2 \vee \dots \vee F_2 \setminus asp_k, \dots, \\ F_n^* &= F_n \setminus asp_1 \vee F_n \setminus asp_2 \vee \dots \vee F_n \setminus asp_k \quad (7). \end{aligned}$$

*Выполнив замену фактов в исходном наборе продукционных правил сформированными на предыдущем шаге аспектами, с учётом обновленного набора фактов, получаем множество антецедентов, состоящее из пары «множество фактов, для которых не выявлен аспект, множество аспектов, входящих в антецедент»:*

$$\begin{aligned} ASP_1^* &= \{asp_1 \in ant_1 \vee asp_2 \in ant_1 \vee \dots \vee asp_k \in ant_1\}, \\ ASP_2^* &= \{asp_1 \in ant_2 \vee asp_2 \in ant_2 \vee \dots \vee asp_k \in ant_2\}, \dots, \\ ASP_n^* &= \{asp_1 \in ant_n \vee asp_2 \in ant_n \vee \dots \vee asp_k \in ant_n\} \quad (8), \end{aligned}$$

$$\begin{aligned} ANT^* &= \{ant_1^* = \langle F_1^*, ASP_1^* \rangle, \\ &ant_2^* = \langle F_2^*, ASP_2^* \rangle, \dots, \\ &ant_n^* = \langle F_n^*, ASP_n^* \rangle\} \quad (9), \end{aligned}$$

где  $ASP_1^*, ASP_2^*, \dots, ASP_n^*$  – множество аспектов, принадлежащих антецеденту.

7. Вернуть модифицированный путём внедрения аспектов набор продукционных правил:

*Следовательно, БЗ будет представлена следующим множеством правил:*

$$R^* = \{rule_1^* \in ant_1^*, \\ rule_2^* \in ant_2^*, \dots, \\ rule_n^* \in ant_n^*\} \text{ (10).}$$

### **2.3.3. Рекурсивный алгоритм извлечения аспектов из последовательности фактов**

Рекурсивный алгоритм извлечения аспектов из последовательности фактов (РАИ-алгоритм), базирующийся на ОАИ-алгоритме, позволяет упростить правила БЗ путём выделения аспектов и мета-аспектов.

Особенности полученных в результате применения РАИ-алгоритма правил:

- меньшее количество сквозных фактов, по сравнению с исходным набором правил и с набором правил, полученным путём применения ОАИ-алгоритма;
- высокий уровень абстракции.

Описание РАИ-алгоритма:

1. Выполнить шаги 1 – 7 ОАИ-алгоритма:

*В результате БЗ будет представлена множеством правил  $R^*$ .*

2. Выполнить поиск мета-аспектов среди отдельных фактов и аспектов:

*Согласно (9) каждый антецедент, состоит из пары «множество фактов, для которых не выявлен аспект, множество аспектов, входящих в антецедент».*

*Мета-аспектом  $masp_j$  будем называть пересечение пар отдельных фактов и аспектов или аспектов в независимых антецедентах, т.е.:*

$$masp_j = \langle F_1^*, ASP_1^* \rangle \cap \langle F_2^*, ASP_2^* \rangle \vee \\ \langle F_1^*, ASP_1^* \rangle \cap \langle F_3^*, ASP_3^* \rangle \vee \\ \langle F_1^*, ASP_1^* \rangle \cap \langle F_4^*, ASP_4^* \rangle \vee \\ \langle F_2^*, ASP_2^* \rangle \cap \langle F_3^*, ASP_3^* \rangle \vee \\ \langle F_2^*, ASP_2^* \rangle \cap \langle F_4^*, ASP_4^* \rangle \vee \dots \vee$$

$$\langle F_{n-1}^*, ASP_{n-1}^* \rangle \cap \langle F_n^*, ASP_n^* \rangle \text{ (11),}$$

где  $j \in N$  – номер выявленного мета-аспекта.

3. Исходя из результатов поиска оформить набор мета-аспектов:

Множество мета-аспектов представлено следующим образом:

$$MASP = \{masp_1, masp_2, \dots, masp_l\} \text{ (12),}$$

где  $l \in N$  – количество выявленных мета-аспектов.

4. Выполнить замену фактов в наборе продукционных правил сформированными на предыдущем шаге мета-аспектами:

Каждое множество фактов и каждый аспект отдельных антецедентов, в случае выявления мета-аспекта, уменьшится на количество фактов и аспектов или аспектов, входящих в мета-аспект:

$$\begin{aligned} F_1^{**} &= F_1^* \setminus masp_1 \vee F_1^* \setminus masp_2 \vee \dots \vee F_1^* \setminus masp_l, \\ F_2^{**} &= F_2^* \setminus masp_1 \vee F_2^* \setminus masp_2 \vee \dots \vee F_2^* \setminus masp_l, \dots, \\ F_n^{**} &= F_n^* \setminus masp_1 \vee F_n^* \setminus masp_2 \vee \dots \vee F_n^* \setminus masp_l, \\ ASP_1^{**} &= ASP_1^* \setminus masp_1 \vee ASP_1^* \setminus masp_2 \vee \dots \vee ASP_1^* \setminus masp_l, \\ ASP_2^{**} &= ASP_2^* \setminus masp_1 \vee ASP_2^* \setminus masp_2 \vee \dots \vee ASP_2^* \setminus masp_l, \dots, \\ ASP_n^{**} &= ASP_n^* \setminus masp_1 \vee ASP_n^* \setminus masp_2 \vee \dots \vee ASP_n^* \setminus masp_l \text{ (13).} \end{aligned}$$

Выполнив замену фактов и аспектов в наборе продукционных правил сформированными на предыдущем шаге мета-аспектами, с учётом обновленного набора фактов и аспектов, получаем множество антецедентов, состоящее из тройки «множество фактов, для которых не выявлен аспект или мета-аспект, множество аспектов, входящих в антецедент, множество мета-аспектов, входящих в антецедент»:

$$\begin{aligned} MASP_1^* &= \{masp_1 \in ant_1 \vee masp_2 \in ant_1 \vee \dots \vee masp_l \in ant_1\}, \\ MASP_2^* &= \{masp_1 \in ant_2 \vee masp_2 \in ant_2 \vee \dots \vee masp_l \in ant_2\}, \dots, \\ MASP_n^* &= \{masp_1 \in ant_n \vee masp_2 \in ant_n \vee \dots \vee masp_l \in ant_n\} \text{ (14),} \\ ANT^{**} &= \{ant_1^{**} = \langle F_1^{**}, ASP_1^{**}, MASP_1^* \rangle, \\ &ant_2^{**} = \langle F_2^{**}, ASP_2^{**}, MASP_2^* \rangle, \dots, \\ &ant_n^{**} = \langle F_n^{**}, ASP_n^{**}, MASP_n^* \rangle\} \text{ (15),} \end{aligned}$$

где  $MASP_1^*$ ,  $MASP_2^*$ , ...,  $MASP_n^*$  – множество мета-аспектов, принадлежащих антецеденту.

5. Вернуть модифицированный путём внедрения мета-аспектов набор продукционных правил:

Следовательно, БЗ будет представлена следующим множеством правил:

$$R^{**} = \{rule_1^{**} \in ant_1^{**}, \\ rule_2^{**} \in ant_2^{**}, \dots, \\ rule_n^{**} \in ant_n^{**}\} \text{ (16).}$$

6. Для повышения абстракции повторять шаги 1 – 5, используя множество правил, обновленное внедрением мета-аспектов, полученных после выполнения данных шагов. Повторять до тех пор, пока не будут выявлены все мета-аспекты.

7. Вернуть модифицированный путём рекурсивного внедрения мета-аспектов набор продукционных правил.

#### 2.4. Выводы по главе

В данной главе были представлены общеизвестные методы решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП:

- фильтры;
- фреймы;
- дуальные экспертные системы;
- нейронные сети;
- генетические алгоритмы;
- Rete-алгоритм.

Было дано объяснение невозможности применимости данных методов при сертификации продукционных экспертных ИС встраиваемого ПО АСУ ТП.

Было представлено описание разработанных алгоритмов решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП, базирующихся на применении АО-подхода:

- одноитерационный алгоритм извлечения аспектов из последовательности фактов;
- рекурсивный алгоритм извлечения аспектов из последовательности фактов.

### Глава 3. АНАЛИЗ КРИТЕРИЕВ ПОВЫШЕНИЯ СКОРОСТИ ВЫЧИСЛИТЕЛЬНЫХ ОПЕРАЦИЙ В ПРОМЫШЛЕННЫХ ПРОДУКЦИОННЫХ ЭКСПЕРТНЫХ ИС

#### 3.1. Исходные данные

Рассмотрим продукционную экспертную ИС выбора мер защиты информации в АСУ ТП, предъявляемых для соответствующего класса защищенности АСУ ТП в соответствии с требованиями приказа Федеральной службы по техническому и экспортному контролю от 14 марта 2014 г. № 31 [63], применяемую для интеллектуальной поддержки специалистов испытательной лаборатории ЗАО ГИЦ ПС ВТ при сертификации промышленных ИС.

Выделенные, согласно [63], требования, представлены в виде объектов правил (антецедентов) продукции и объектов действий (консеквентов) продукции, и приведены в Приложении В.

Фрагмент набора продукционных правил, с учетом выделенных объектов правил и объектов действий, представлен в листинге 1.

```
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action1;
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action2;
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action3;
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action4;
...
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr14 ТО action135;
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr14 ТО action136;
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr14 ТО action137;
...
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action101;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action102;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action104;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action105;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action106;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action108;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action109;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr11 ТО action110;
...
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr17 ТО action151;
ЕСЛИ fact1=fact1attr2 И fact2=fact2attr17 ТО action152;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action1;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action2;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action3;
```

```

ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action4;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action5;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action6;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr1 ТО action8;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr2 ТО action9;
...
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action109;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action110;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action112;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action113;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action114;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action115;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action116;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action119;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr11 ТО action120;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action121;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action122;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action123;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action124;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action125;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action126;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr12 ТО action127;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr13 ТО action128;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr13 ТО action130;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr13 ТО action131;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr14 ТО action133;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr14 ТО action134;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr14 ТО action135;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr14 ТО action136;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr14 ТО action137;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr15 ТО action138;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr15 ТО action139;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr15 ТО action140;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action141;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action142;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action143;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action144;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action145;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action146;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr16 ТО action147;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr17 ТО action148;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr17 ТО action149;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr17 ТО action150;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr17 ТО action151;
ЕСЛИ fact1=fact1attr3 И fact2=fact2attr17 ТО action152;

```

Листинг 1. Фрагмент набора продукционных правил

Граф логического вывода для данного (исходного) набора продукционных правил представлен на рисунке 10.

**Выбор мер защиты информации в АСУ ТП, предъявляемых для соответствующего класса защищенности АСУ ТП**

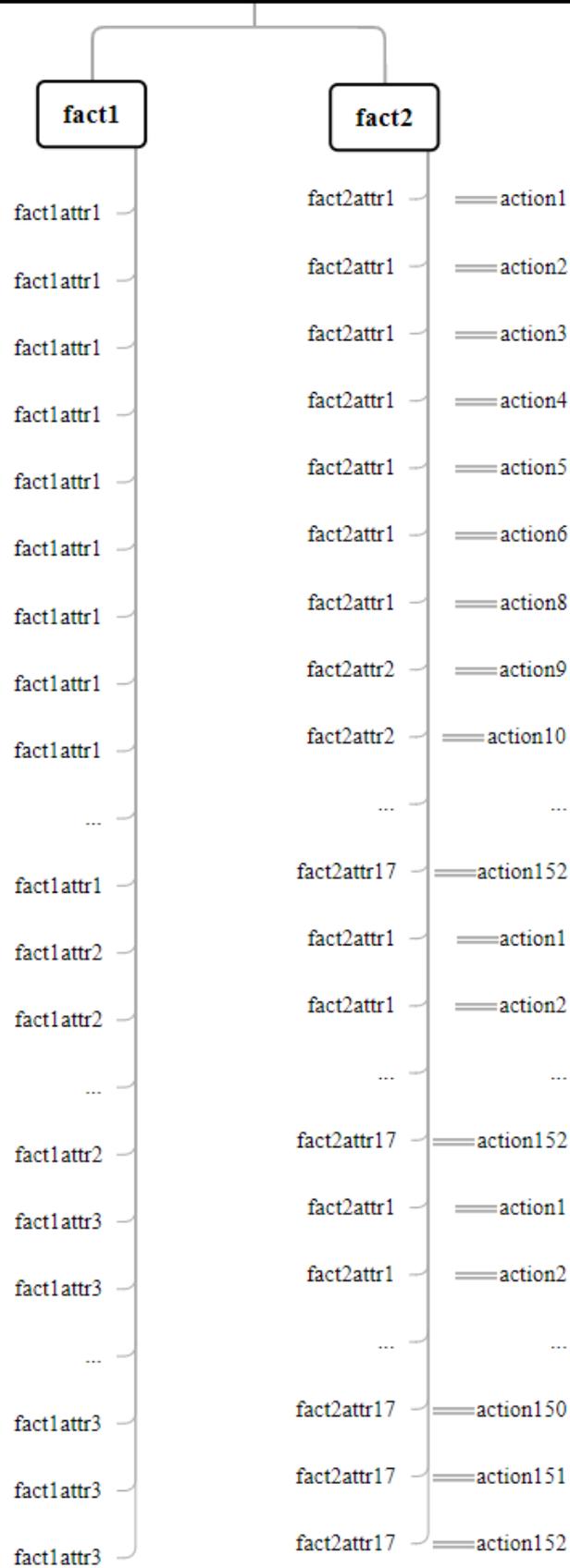


Рисунок 10. Граф логического вывода исходного набора продукционных правил

На рисунке 11 зафиксированы значения скорости логического вывода в БЗ экспертной ИС из 322 правил [22]. Поиск в БЗ проводился без объяснения решения. На оси у отмечено время выполнения поиска решения в микросекундах, на оси x отмечены номера проведенных тестов.

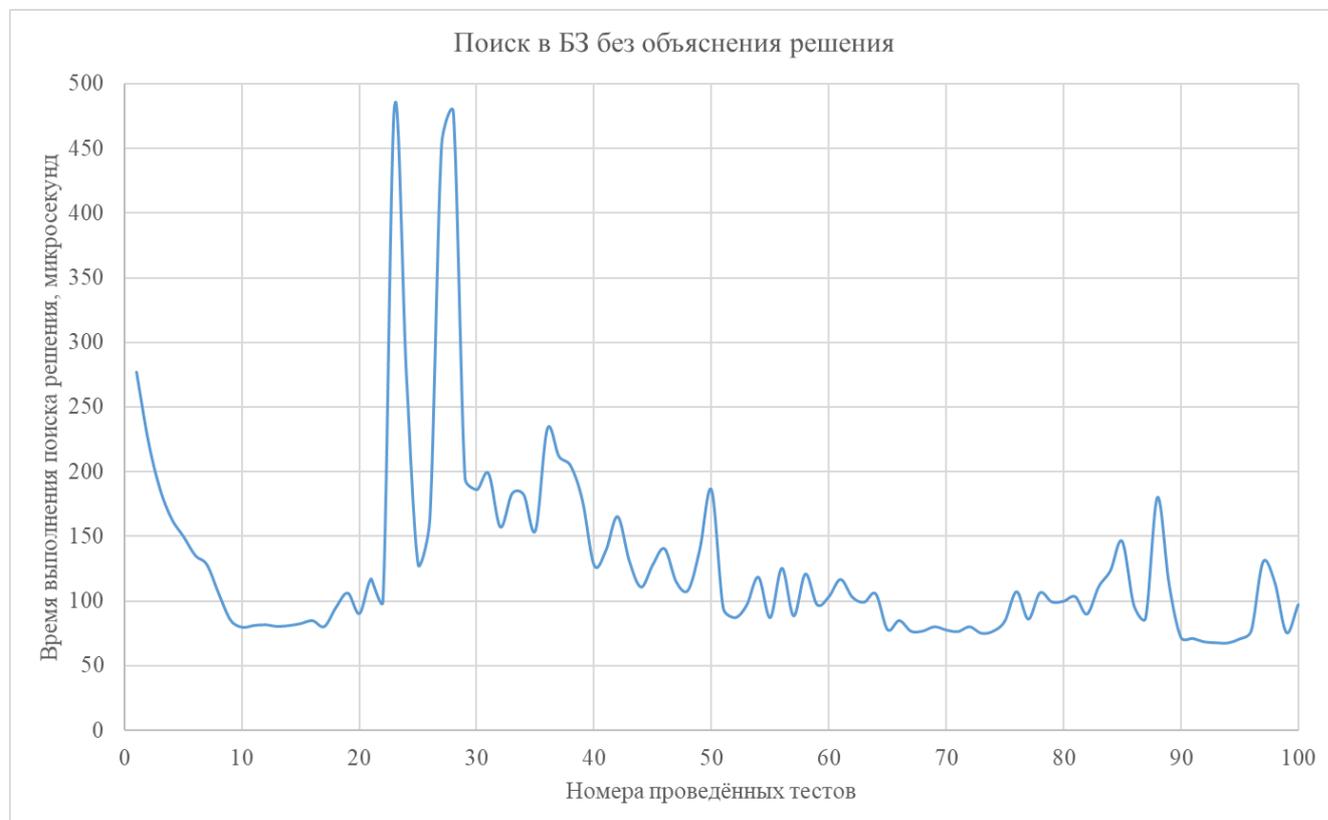


Рисунок 11. Зафиксированное значение скорости поиска в БЗ без объяснения решения

### 3.2. Анализ критериев повышения скорости вычислительных операций в продукционных экспертных ИС

В качестве критериев повышения скорости вычислительных операций экспертами (D. Tarditi, G. Morrisett, P. Cheng, C. Stone, R. Harper, P. Lee [97], D. Volpano, G. Smith [99]) выделяются:

- уменьшение занимаемого вычислительным процессом объема памяти;
- совершенствование алгоритмов выполнения вычислительных операций;
- уменьшение задержек при передаче информации;
- устранение дублирования информации (нормализация);
- распараллеливание вычислительных процессов.

Исходя из приведённых критериев на скорость логического вывода в БЗ экспертной ИС в большей мере влияют критерии, приведённые в таблице 4.

Таблица 4. Основные критерии, влияющие на скорость логического вывода в БЗ экспертной ИС

№ п/п	Критерий	Описание
1	Скорость формирования	Скорость формирования продукционных правил
2	Занимаемый объем памяти	Объем памяти занимаемый массивом продукционных правил
3	Затраты времени	Затраты времени на логический вывод

Данные критерии будут учитываться в МАИ для определения лучшей из альтернатив.

Для повышения эффективности процесса логического вывода в продукционных экспертных ИС ранее были разработаны алгоритмы, базирующиеся на применении АО-подхода. В следующем подразделе будут представлены результаты использования данных алгоритмов на исходном наборе продукционных правил.

### 3.3. Результаты применения разработанных алгоритмов решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС

#### 3.3.1. Одноитерационный алгоритм извлечения аспектов из последовательности фактов

В листинге 2 представлен набор продукционных правил, согласно требованиям, представленным в [63], с учетом выделенных объектов правил и объектов действий, после применения ОАИ-алгоритма.

```
ЕСЛИ aspect1 ТО action1;  
ЕСЛИ aspect1 ТО action2;  
ЕСЛИ aspect1 ТО action3;  
ЕСЛИ aspect1 ТО action4;  
ЕСЛИ aspect1 ТО action5;  
ЕСЛИ aspect1 ТО action6;  
ЕСЛИ aspect1 ТО action8;  
ЕСЛИ aspect2 ТО action9;  
ЕСЛИ aspect2 ТО action10;  
ЕСЛИ aspect2 ТО action11;  
ЕСЛИ aspect2 ТО action13;
```

```

ЕСЛИ aspect2 ТО action14;
ЕСЛИ aspect2 ТО action15;
ЕСЛИ aspect2 ТО action19;
ЕСЛИ aspect2 ТО action20;
ЕСЛИ aspect2 ТО action22;
ЕСЛИ aspect2 ТО action23;
ЕСЛИ aspect3 ТО action28;
ЕСЛИ aspect3 ТО action29;
ЕСЛИ aspect3 ТО action30;
ЕСЛИ aspect3 ТО action33;
ЕСЛИ aspect3 ТО action35;
ЕСЛИ aspect3 ТО action36;
ЕСЛИ aspect4 ТО action37;
ЕСЛИ aspect4 ТО action38;
ЕСЛИ aspect4 ТО action39;
ЕСЛИ aspect4 ТО action40;
ЕСЛИ aspect4 ТО action41;
ЕСЛИ aspect4 ТО action43;
ЕСЛИ aspect4 ТО action44;
ЕСЛИ aspect4 ТО action45;
ЕСЛИ aspect4 ТО action47;
ЕСЛИ aspect5 ТО action49;
ЕСЛИ aspect5 ТО action50;
ЕСЛИ aspect5 ТО action51;
ЕСЛИ aspect5 ТО action53;
...
aspect1 = [fact1=fact1attr1 И fact2=fact2attr1]
aspect2 = [fact1=fact1attr1 И fact2=fact2attr2];
aspect3 = [fact1=fact1attr1 И fact2=fact2attr4];
aspect4 = [fact1=fact1attr1 И fact2=fact2attr5];
aspect5 = [fact1=fact1attr1 И fact2=fact2attr6];
...

```

Листинг 2. Фрагмент набора продукционных правил после применения ОАИ-алгоритма

Граф логического вывода для набора продукционных правил после применения ОАИ-алгоритма представлен на рисунке 12.

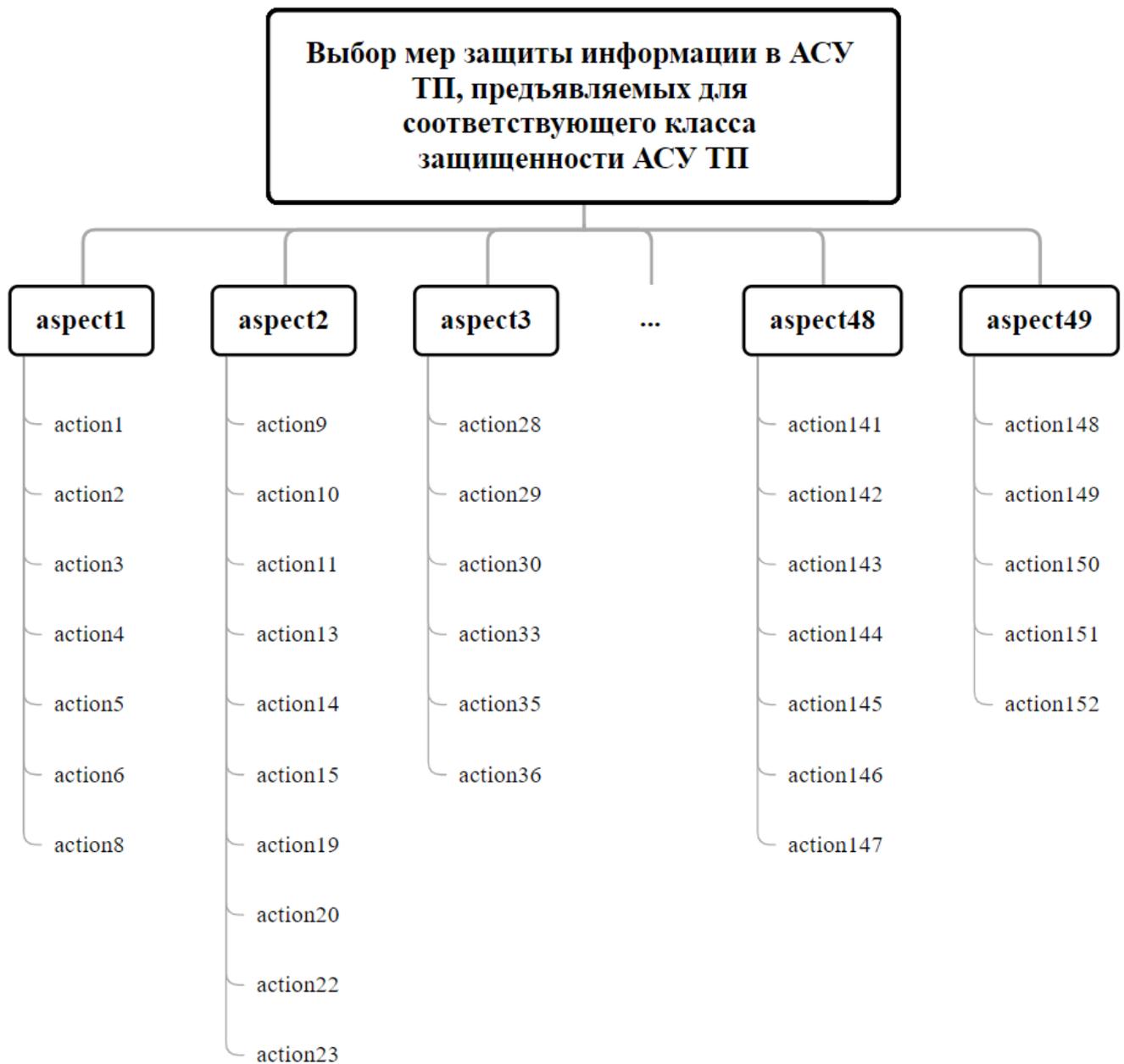


Рисунок 12. Граф логического вывода для набора продукционных правил после применения ОАИ-алгоритма

### 3.3.2. Рекурсивный алгоритм извлечения аспектов из последовательности фактов

В листинге 3 представлен набор продукционных правил, согласно требованиям, представленным в [63], с учетом выделенных объектов правил и объектов действий, после применения РАИ-алгоритма.

```

ЕСЛИ aspect50 ТО action1 И action2 И action3 И action4 И action5
И action6 И action8;
ЕСЛИ aspect51 ТО action9 И action10 И action11 И action13 И
action14 И action15 И action19 И action20 И action22 И action23;
  
```

```

ЕСЛИ aspect52 ТО action28 И action29 И action30 И action33 И
action35 И action36;
ЕСЛИ aspect53 ТО action37 И action38 И action39 И action40 И
action41 И action43 И action44 И action45 И action47;
ЕСЛИ aspect54 ТО action49 И action50 И action51 И action53;
ЕСЛИ aspect55 ТО action58 И action59;
...
aspect1 = [fact1=fact1attr1 И fact2=fact2attr1]
aspect2 = [fact1=fact1attr1 И fact2=fact2attr2];
aspect3 = [fact1=fact1attr1 И fact2=fact2attr4];
aspect4 = [fact1=fact1attr1 И fact2=fact2attr5];
aspect5 = [fact1=fact1attr1 И fact2=fact2attr6];
...
aspect50 = _ [aspect1];
aspect51 = _ [aspect2];
aspect52 = _ [aspect3];
aspect53 = _ [aspect4];
aspect54 = _ [aspect5];
...
aspect73 = _ [aspect24];
aspect74 = _ [aspect25];
aspect75 = _ [aspect26];
aspect76 = _ [aspect27];
aspect77 = _ [aspect28];
aspect78 = _ [aspect29];
aspect79 = _ [aspect30];
aspect80 = _ [aspect31];
aspect81 = _ [aspect32];
aspect82 = _ [aspect33];
aspect83 = _ [aspect34];
aspect84 = _ [aspect35];
aspect85 = _ [aspect36];
aspect86 = _ [aspect37];
aspect87 = _ [aspect38];
aspect88 = _ [aspect39];
aspect89 = _ [aspect40];
aspect90 = _ [aspect41];
aspect91 = _ [aspect42];
aspect92 = _ [aspect43];
aspect93 = _ [aspect44];
aspect94 = _ [aspect45];
aspect95 = _ [aspect46];
aspect96 = _ [aspect47];
aspect97 = _ [aspect48];
aspect98 = _ [aspect49];

```

Листинг 3. Фрагмент набора продукционных правил после применения РАИ-алгоритма

Граф логического вывода для набора продукционных правил после применения РАИ-алгоритма представлен на рисунке 13.

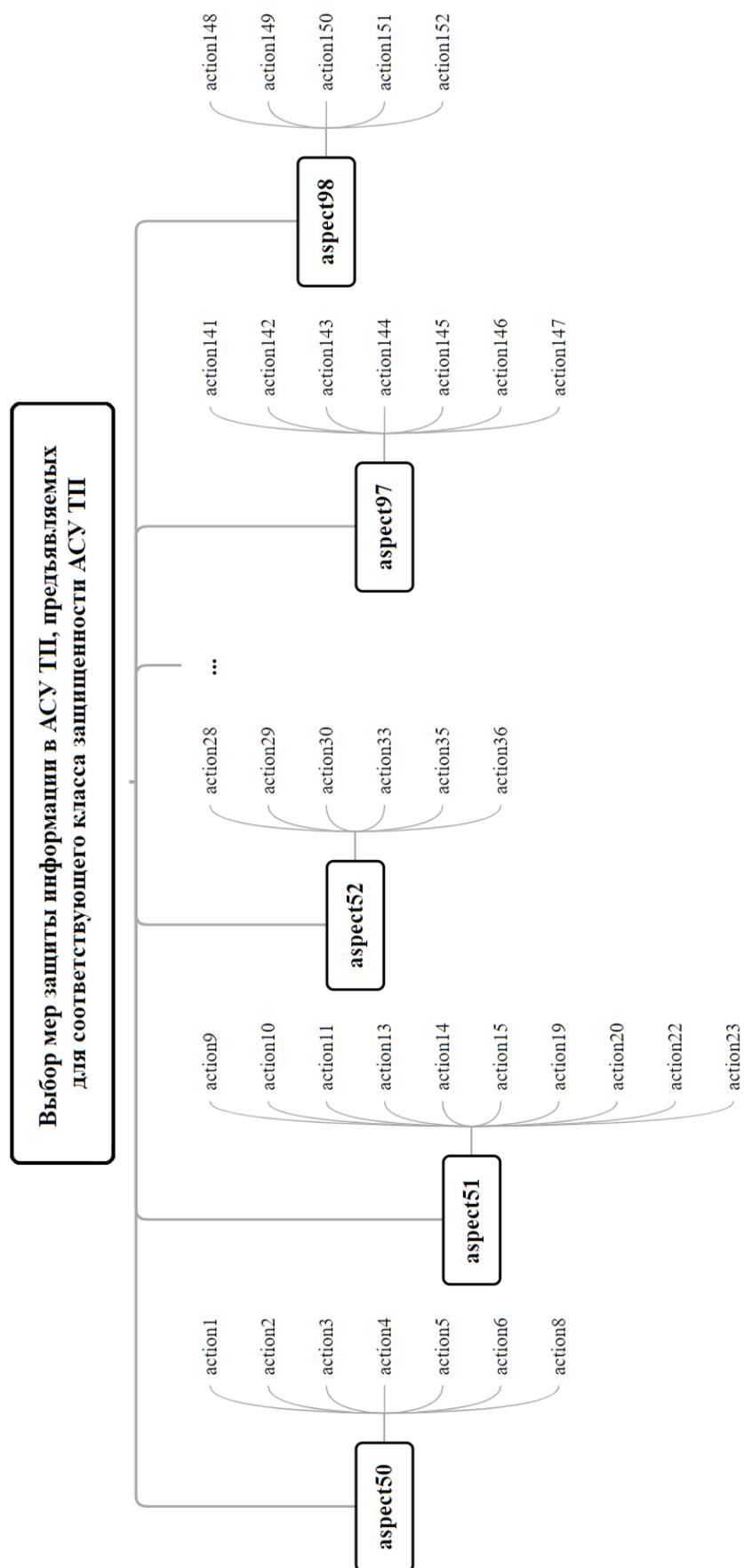


Рисунок 13. Граф логического вывода для набора продукционных правил после применения РАИ-алгоритма

### 3.4. Использование МАИ для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в базе знаний экспертной ИС

Для поддержки принятия решения при выборе алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС будет использован МАИ.

МАИ был разработан в начале 1970 года американским математиком Томасом Саати [Thomas L. Saaty]. Данный метод позволяет выразить проблему выбора в виде иерархии. В наиболее распространенном случае иерархия состоит из вершины – цели, промежуточного уровня – критериев (техничко-экономических параметров) и нижнего уровня – набора альтернатив.

Иерархическое воспроизведение проблемы позволяет установить приоритеты критериев и оценить каждую из альтернатив согласно критериям. МАИ позволяет выполнить попарное (относительно воздействия на совместную (общую) их характеристику) сравнение элементов задачи.

Результат парных сравнений образует обратносимметричную матрицу. Элементом матрицы  $a(i, j)$  является интенсивность проявления элемента иерархии  $i$  относительно элемента иерархии  $j$ .

Для фиксации интенсивности используется шкала отношений Саати:

1 – равное соотношение (важность);

3 – умеренное превосходство (одного над другим);

5 – существенное превосходство;

7 – значительное превосходство;

9 – очень сильное превосходство;

2, 4, 6, 8 – промежуточное решение между двумя соседними суждениями.

В МАИ элементы  $a(i, j)$  должны соответствовать следующим правилам:

– если при сравнении одного фактора  $i$  со вторым фактором  $j$  получено  $a(i, j) = b$ , то при сравнении второго фактора с первым получаем  $a(j, i) = 1/b$ .

– для всех  $i$  значение элемента матрицы  $a(i, j) = 1$ .

Согласно описанным правилам матрица парных сравнений  $\bar{A}$  имеет вид:

$$\bar{A} = \begin{pmatrix} 1 & A_1 & \dots & A_n \\ A_1 & 1 & \dots & W_1/W_n \\ \dots & \dots & 1 & A_n \\ A_n & W_n/W_1 & \dots & 1 \end{pmatrix} \quad (17),$$

где  $A_1, A_2, \dots, A_n$  – множество элементов матрицы парных сравнений,  $W_1, W_2, \dots, W_n$  – интенсивность проявления элемента.

Оценка компонентов вектора приоритетов производится по следующим формулам:

Собственный вектор:

$$\bar{A} = \begin{pmatrix} 1 & A_1 & \dots & A_n \\ A_1 & 1 & \dots & \frac{W_1}{W_n} \\ \dots & \dots & 1 & A_n \\ A_n & \frac{W_n}{W_1} & \dots & 1 \end{pmatrix}$$

$$X_1 = \sqrt[n]{1 * W_1/W_2 * \dots * W_1/W_n} \quad (18).$$

$$X_n = \sqrt[n]{\dots * W_n/W_1 * \dots * 1}$$

Вектор приоритетов:

$$\bar{A} = \begin{pmatrix} 1 & A_1 & \dots & A_n \\ A_1 & 1 & \dots & \frac{W_1}{W_n} \\ \dots & \dots & 1 & A_n \\ A_n & \frac{W_n}{W_1} & \dots & 1 \end{pmatrix}$$

$$X_1 / \sum_{i=1}^n X_i \quad (19).$$

$$\dots \\ X_n / \sum_{i=1}^n X_i$$

Далее вычисляются максимальное собственное значение  $\lambda_{max}$ , индекс согласованности ИС и отношение согласованности ОС.

Максимальное собственное значение  $\lambda_{max}$  вычисляется как сумма произведений суммы столбцов матрицы попарных сравнений на величину соответствующего вектора приоритетов.

Индекс согласованности ИС необходим для проверки степени согласованности и вычисляется по формуле:

$$ИС = (\lambda_{max} - n) / (n - 1) \quad (20).$$

Отношение согласованности ОС определяется из отношения индекса согласованности и числа, которое соответствует значению случайной согласованности СС матрицы того же порядка, что и исходная матрица попарных сравнений:

$$ОС = ИС / СС \quad (21).$$

На текущий момент для шкалы учеными Национальной лаборатории Окриджа на базе 100 случайных выборок [71] рассчитаны случайные индексы согласованности для обратно симметричных матриц размерностью от 1 до 15.

Значение показателя случайной согласованности представлено в таблице 5.

Таблица 5 – Значение показателя случайной согласованности

<b>Размер матрицы</b>	<b>Значение показателя случайной согласованности</b>
1	0.00
2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
15	1.59

Величина ОС должна быть порядка 10 % или менее. Иначе величина ОС не является приемлемой и необходимо перепроверить суждения.

Значение глобального приоритета для каждой из альтернатив подсчитывается как сумма произведений значения вектора приоритета для критерия и значений вектора локального приоритета этой альтернативы в отношении данного критерия.

Таким образом, МАИ включает в себя следующие этапы:

Метод состоит из совокупности следующих этапов:

1. Структуризация задачи в виде иерархической структуры с несколькими уровнями;
2. Выполнение попарных сравнений критериев;
3. Вычисление коэффициентов важности для критериев (при этом проверяется согласованность суждений);
4. Выполнение попарных сравнений альтернатив;
5. Вычисление коэффициентов важности для альтернатив (при этом проверяется согласованность суждений);
6. Определение наилучшей альтернативы.

Предложенный подход дает объективную оценку при выборе алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС.

Применим МАИ для поддержки принятия решения при выборе алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС: 10, 25, 50, 100, 250, 500 и 1000 продукционных правил.

Рассмотрим МАИ для БЗ экспертной ИС состоящей из 10 продукционных правил.

### **3.4.1. Структуризация задачи в виде иерархической структуры с несколькими уровнями**

Связь между целью, критериями оценивания и альтернативами образуют иерархическую структуру. Целью является выбор алгоритма извлечения аспектов

из последовательности фактов, альтернативами – разработанные алгоритмы, а критериями оценивания – выбранные на первом этапе анализа характеристики: скорость формирования, занимаемый объем памяти, затраты времени.

Иерархическая структура воспроизведения проблемы выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС встраиваемого ПО АСУ ТП представлена на рисунке 14.

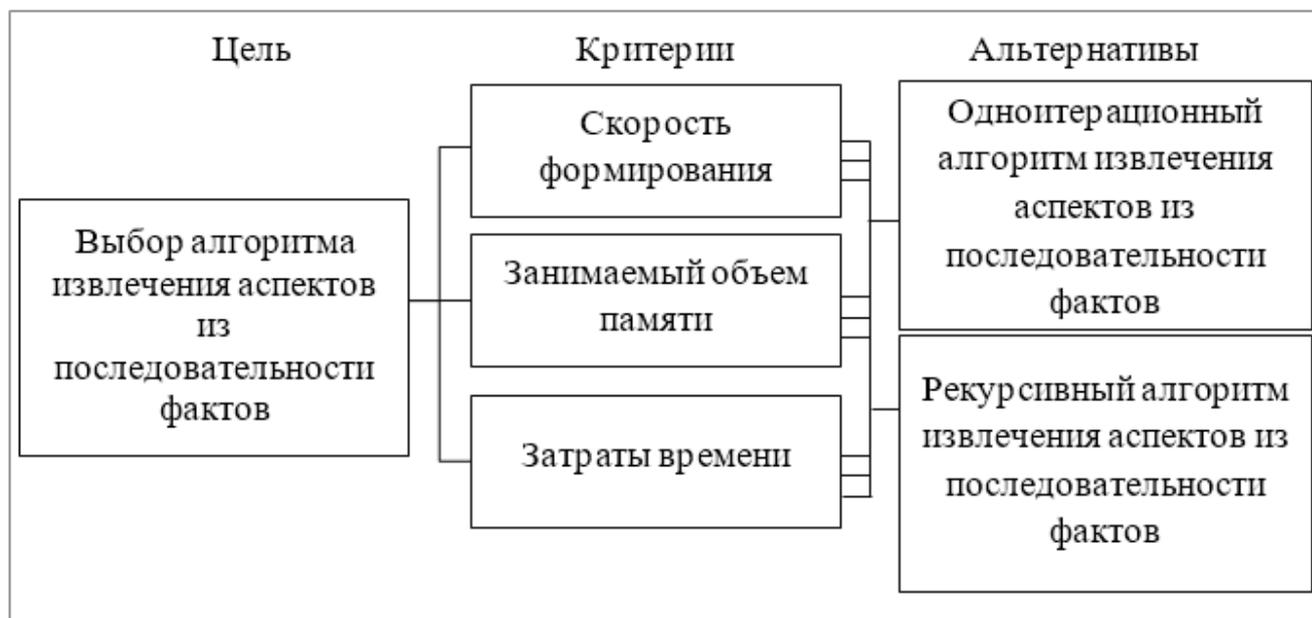


Рисунок 14. Иерархическая структура воспроизведения проблемы выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС встраиваемого ПО АСУ ТП

### 3.4.2. Выполнение попарных сравнений критериев

Результаты сравнения пары критериев представлены в таблице 6.

Таблица 6. Матрица попарных сравнений критериев

<i>Критерий</i>	Скорость формирования	Занимаемый объем памяти	Затраты времени
Скорость формирования	1/1	2/1	5/1
Занимаемый объем памяти	1/2	1/1	4/1
Затраты времени	1/5	1/4	1/1

### 3.4.3. Вычисление коэффициентов важности для критериев

Полученные согласно формулам (18) и (19) значения собственных векторов и векторов приоритетов для критериев представлены в таблице 7.

Таблица 7. Значения собственных векторов и векторов приоритетов для критериев

<i>Критерий</i>	Скорость формирования	Занимаемый объем памяти	Затраты времени	<i>Собственный вектор</i>	<i>Вектор приоритетов</i>
Скорость формирования	1,000	2,000	5,000	<b>2,154</b>	<b>0,570</b>
Занимаемый объем памяти	0,500	1,000	4,000	<b>1,260</b>	<b>0,333</b>
Затраты времени	0,200	0,250	1,000	<b>0,368</b>	<b>0,097</b>
<b>Итого</b>	<b>1,700</b>	<b>3,250</b>	<b>10,000</b>	<b>3,783</b>	<b>1,000</b>

Сумма значений векторов приоритетов равна 1, значит нормализация (составление векторов приоритетов) выполнена корректно.

Определим величину отношения согласованности ОС, используя формулы (20) и (21).

$$\lambda_{max} = 0,570 * 1,700 + 0,333 * 3,250 + 0,097 * 10,000 = 3,025,$$

$$ИС = (3,025 - 3) / (3 - 1) = 0,012, ОС = 0,012 / 0,580 = 0,021 = 2,1\%,$$

где 0,580 – значение случайной согласованности ОС для  $n = 3$ .

Величина ОС = 2,1 % < 10 %, что является приемлемым результатом, значит перепроверять суждения не нужно.

Выполнив сравнение нормализованных оценок векторов приоритетов, приходим к выводу – для 10 продукционных правил наибольшее значение при выборе алгоритма извлечения аспектов из последовательности фактов придается критерию «Скорость формирования».

### 3.4.4. Выполнение попарных сравнений альтернатив

Результаты сравнения пары альтернатив для критерия «Скорость формирования» представлены в таблице 8. Результаты сравнения пары альтернатив для критерия «Занимаемый объём памяти» представлены в таблице 9. Результаты сравнения пары альтернатив для критерия «Затраты времени» представлены в таблице 10.

Таблица 8. Матрица попарных сравнений альтернатив для критерия «Скорость формирования»

<i>«Скорость формирования»</i>	Без применения алгоритма	ОАИ-алгоритм	РАИ-алгоритм
Без применения алгоритма	1/1	3/1	4/1
ОАИ-алгоритм	1/3	1/1	2/1
РАИ-алгоритм	1/4	1/2	1/1

Таблица 9. Матрица попарных сравнений альтернатив для критерия «Занимаемый объём памяти»

<i>«Занимаемый объём памяти»</i>	Без применения алгоритма	ОАИ-алгоритм	РАИ-алгоритм
Без применения алгоритма	1/1	1/5	1/5
ОАИ-алгоритм	5/1	1/1	2/1
РАИ-алгоритм	5/1	1/2	1/1

Таблица 10. Матрица попарных сравнений альтернатив для критерия «Затраты времени»

<i>«Затраты времени»</i>	Без применения алгоритма	ОАИ-алгоритм	РАИ-алгоритм
Без применения алгоритма	1/1	1/8	1/7
ОАИ-алгоритм	8/1	1/1	3/1
РАИ-алгоритм	7/1	1/3	1/1

### 3.4.5. Вычисление коэффициентов важности для альтернатив

Полученные согласно формулам (18) и (19) значения собственных векторов и векторов приоритетов альтернатив для критерия «Скорость формирования»

представлены в таблице 11, для критерия «Занимаемый объём памяти» представлены в таблице 12, для критерия «Затраты времени» представлены в таблице 13.

Таблица 11. Значения собственных векторов и векторов приоритетов альтернатив для критерия «Скорость формирования»

<b>«Скорость формирования»</b>	Без применения алгоритма	ОАИ-алгоритм	РАИ-алгоритм	<b>Собственный вектор</b>	<b>Вектор приоритетов</b>
Без применения алгоритма	1,000	3,000	4,000	<b>2,289</b>	<b>0,625</b>
ОАИ-алгоритм	0,333	1,000	2,000	<b>0,874</b>	<b>0,238</b>
РАИ-алгоритм	0,250	0,500	1,000	<b>0,500</b>	<b>0,136</b>
<b>Итого</b>	<b>1,580</b>	<b>4,500</b>	<b>7,000</b>	<b>3,663</b>	<b>1,000</b>

Сумма значений векторов приоритетов равна 1, значит нормализация (составление векторов приоритетов) выполнена корректно.

Определим величину отношения согласованности ОС, используя формулы (20) и (21).

$$\lambda_{max} = 0,625 * 1,580 + 0,238 * 4,500 + 0,136 * 7,000 = 3,018,$$

$$ИС = (3,018 - 3) / (3 - 1) = 0,009, ОС = 0,009 / 0,580 = 0,016 = 1,6\%,$$

где 0,580 – значение случайной согласованности ОС для  $n = 3$ .

Величина ОС = 1,6 % < 10 %, что является приемлемым результатом, значит перепроверять суждения не нужно.

Таблица 12. Значения собственных векторов и векторов приоритетов альтернатив для критерия «Занимаемый объём памяти»

<b>«Занимаемый объём памяти»</b>	Без применения алгоритма	ОАИ-алгоритм	РАИ-алгоритм	<b>Собственный вектор</b>	<b>Вектор приоритетов</b>
Без применения алгоритма	1,000	0,200	0,200	<b>0,342</b>	<b>0,089</b>
ОАИ-алгоритм	5,000	1,000	2,000	<b>2,154</b>	<b>0,559</b>
РАИ-алгоритм	5,000	0,500	1,000	<b>1,357</b>	<b>0,352</b>
<b>Итого</b>	<b>11,000</b>	<b>1,700</b>	<b>3,200</b>	<b>3,854</b>	<b>1,000</b>

Сумма значений векторов приоритетов равна 1, значит нормализация (составление векторов приоритетов) выполнена корректно.

Определим величину отношения согласованности ОС, используя формулы (20) и (21).

$$\lambda_{max} = 0,089 * 11,000 + 0,559 * 1,700 + 0,352 * 3,200 = 3,054,$$

$$ИС = (3,054 - 3) / (3 - 1) = 0,027, ОС = 0,027 / 0,580 = 0,046 = 4,6\%,$$

где 0,580 – значение случайной согласованности ОС для  $n = 3$ .

Величина ОС = 4,6 % < 10 %, что является приемлемым результатом, значит перепроверять суждения не нужно.

Таблица 13. Значения собственных векторов и векторов приоритетов альтернатив для критерия «Затраты времени»

<b>«Затраты времени»</b>	<b>Без применения алгоритма</b>	<b>ОАИ-алгоритм</b>	<b>РАИ-алгоритм</b>	<b>Собственный вектор</b>	<b>Вектор приоритетов</b>
Без применения алгоритма	1,000	0,125	0,143	<b>0,261</b>	<b>0,058</b>
ОАИ-алгоритм	8,000	1,000	3,000	<b>2,884</b>	<b>0,645</b>
РАИ-алгоритм	7,000	0,333	1,000	<b>1,326</b>	<b>0,297</b>
<b>Итого</b>	<b>16,000</b>	<b>1,460</b>	<b>4,140</b>	<b>4,472</b>	<b>1,000</b>

Сумма значений векторов приоритетов равна 1, значит нормализация (составление векторов приоритетов) выполнена корректно.

Определим величину отношения согласованности ОС, используя формулы (20) и (21).

$$\lambda_{max} = 0,058 * 16,000 + 0,645 * 1,460 + 0,297 * 4,140 = 3,104,$$

$$ИС = (3,104 - 3) / (3 - 1) = 0,052, ОС = 0,052 / 0,580 = 0,090 = 9,0\%,$$

где 0,580 – значение случайной согласованности ОС для  $n = 3$ .

Величина ОС = 9,0 % < 10 %, что является приемлемым результатом, значит перепроверять суждения не нужно.

### **3.4.6. Определение наилучшей альтернативы**

Значение глобального приоритета для каждой из альтернатив подсчитывается как сумма произведений значения вектора приоритета для критерия и значений вектора локального приоритета этой альтернативы в отношении данного критерия.

Рассчитанное значение глобального приоритета для каждой из альтернатив представлено в таблице 14.

Таблица 14. Определение наилучшей альтернативы

<i>Определение наилучшей альтернативы</i>	<i>Критерии</i>			<i>Глобальные приоритеты</i>
<i>Альтернативы</i>	Скорость формирования	Занимаемый объем памяти	Затраты времени	
	<i>Значение вектора приоритета</i>			
	0,570	0,333	0,097	
Без применения алгоритма	2,289	0,342	0,261	<b>1,443</b>
ОАИ-алгоритм	0,874	2,154	2,884	<b>1,496</b>
РАИ-алгоритм	0,500	1,357	1,326	<b>0,866</b>

Выбранной считается альтернатива с максимальным значением глобального приоритета. Таким образом, для БЗ экспертной ИС состоящей из 10 продукционных правил, необходимо выбрать ОАИ-алгоритм.

Повторив данные вычисления для БЗ экспертной ИС из 25, 50, 100, 250, 500 и 1000 продукционных правил, получим гистограмму, отображающую выбранные значения глобальных приоритетов, представленную на рисунке 15.

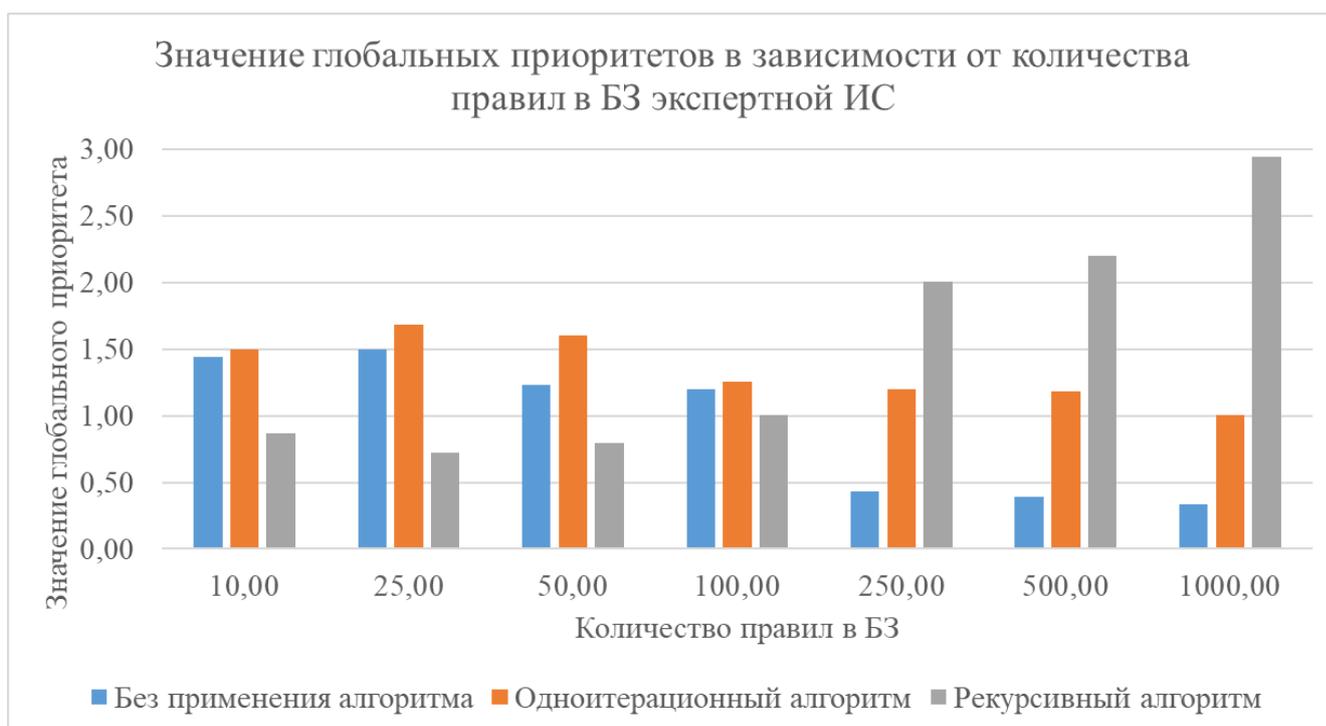


Рисунок 15. Гистограмма, отображающая выбранные значения глобальных приоритетов для БЗ экспертной ИС из 10, 25, 50, 100, 250, 500 и 1000 продукционных правил

Из данной гистограммы видно, что с увеличением количества продукционных правил в БЗ экспертной ИС до 500 правил, лучше всего начинает себя проявлять РАИ-алгоритм.

Результаты, полученные для общего случая, представлены в таблице 15.

Таблица 15. Результаты выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС

№ п/п	Количество продукционных правил	Рекомендуемый алгоритм
1	10	ОАИ-алгоритм
2	25	ОАИ-алгоритм
3	50	ОАИ-алгоритм
4	100	ОАИ-алгоритм
5	250	ОАИ-алгоритм
6	500	РАИ-алгоритм
7	1000	РАИ-алгоритм

На рисунке 16 зафиксированы значения скорости логического вывода в БЗ экспертной ИС из 322 правил после применения, выбранного через МАИ для заданного количества продукционных правил, ОАИ-алгоритма. Поиск в БЗ проводился без объяснения решения. На оси у отмечено время выполнения поиска решения в микросекундах, на оси x отмечены номера проведенных тестов.

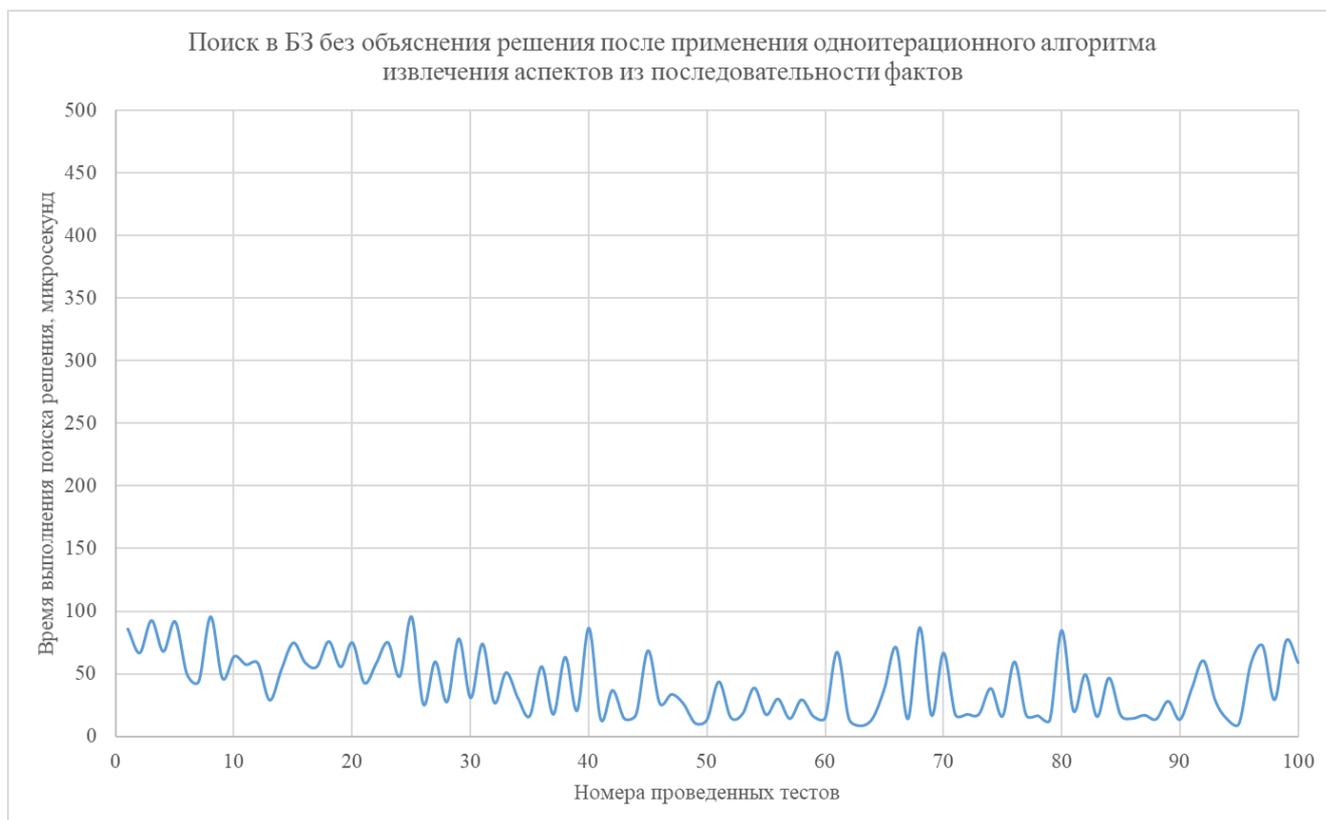


Рисунок 16. Зафиксированное значение скорости поиска в БЗ без объяснения решения после применения ОАИ-алгоритма

Сравнение значений скорости логического вывода в БЗ экспертной ИС из 322 правил до и после применения ОАИ-алгоритма представлено на рисунке 17.

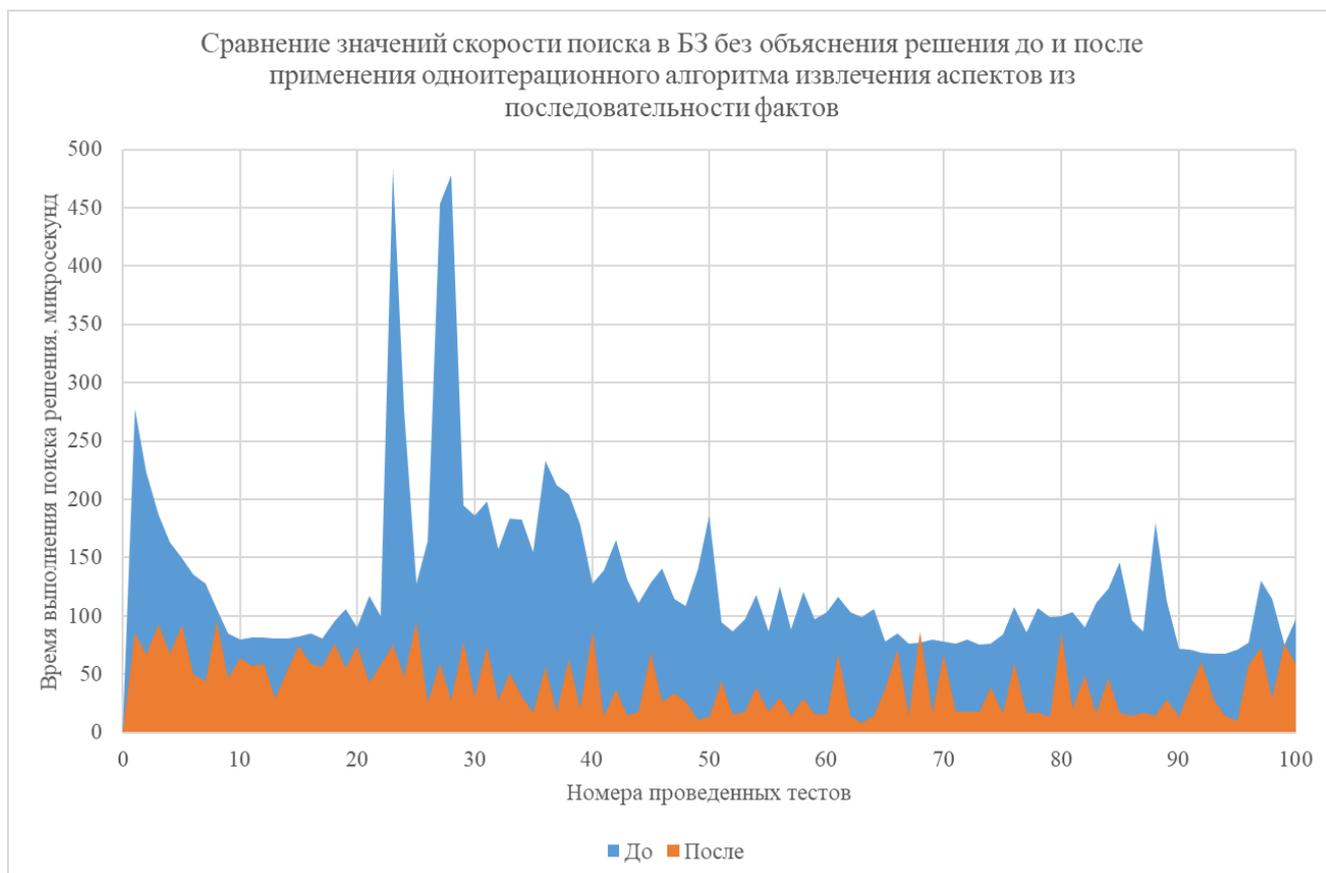


Рисунок 17. Сравнение значений скорости логического вывода в БЗ экспертной ИС из 322 правил до и после применения ОАИ-алгоритма

### 3.5. Выводы по главе

Данная глава была посвящена анализу критериев повышения скорости вычислительных операций в продукционных экспертных ИС.

Были представлены исходные данные и применённые к ним разработанные алгоритмы решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС с использованием АО-подхода.

Приведено описание использования МАИ для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в БЗ экспертной ИС согласно выделенным критериям повышения скорости вычислительных операций в продукционных экспертных ИС.

## Глава 4. ПРОТОТИП ПОДСИСТЕМЫ ИНТЕРПРЕТАЦИИ ПРОДУКЦИОННЫХ ПРАВИЛ ЭКСПЕРТНЫХ ИС С ПРИМЕНЕНИЕМ АО-ПОДХОДА

### 4.1. Изменения модели экспертной ИС в части интерпретации продукционных правил, базирующиеся на применении АО-подхода

Внедрение АО-подхода не требует переработки структуры БЗ и БД экспертной ИС. В целях совместимости предлагается использовать АО-подход в интерпретаторе правил продукционной экспертной ИС.

В соответствии с АО-подходом предлагается следующая последовательность интерпретации продукционных правил:

*Выявленные пересечения фактов или действий в наборе правил образуют аспект. Множество фактов или действий, которые выделены в аспект, образуют срез. Аспект внедряется в правило через антецедент или консеквент. Роль совета в данном подходе играют факт или действие, которые будут заменены элементом аспекта. Правило, факт или действие в котором будут заменены элементом аспекта, обозначается целью. Интерпретатор правил выполняет роль компоновщика.*

Результат анализа составляющих продукционных систем и основных концепций АО-подхода представлен в таблице 16.

Таблица 16. Таблица соответствия компонентов продукционных систем и основных концепций АО-подхода

№ п/п	Концепция АО-подхода	Отражение концепции АО-подхода в продукционных системах
1	Aspect	Выявленные пересечения фактов или действий в наборе правил
2	Pointcut	Множество (срез) фактов или действий, который принадлежит аспекту
3	Join point	Место внедрения элементов аспекта, представленное антецедентом или консеквентом
4	Advice	Факт или действие, которые будут заменены элементом аспекта
5	Target	Правило, факт или действие в котором будут заменены элементом аспекта

№ п/п	Концепция АО-похода	Отражение концепции АО-похода в продукционных системах
6	Weaver	Интерпретатор (компоновщик) правил

На рисунке 18 представлено изменение классической статической модели экспертной ИС в части интерпретации продукционных правил, базирующееся на применении АО-подхода.

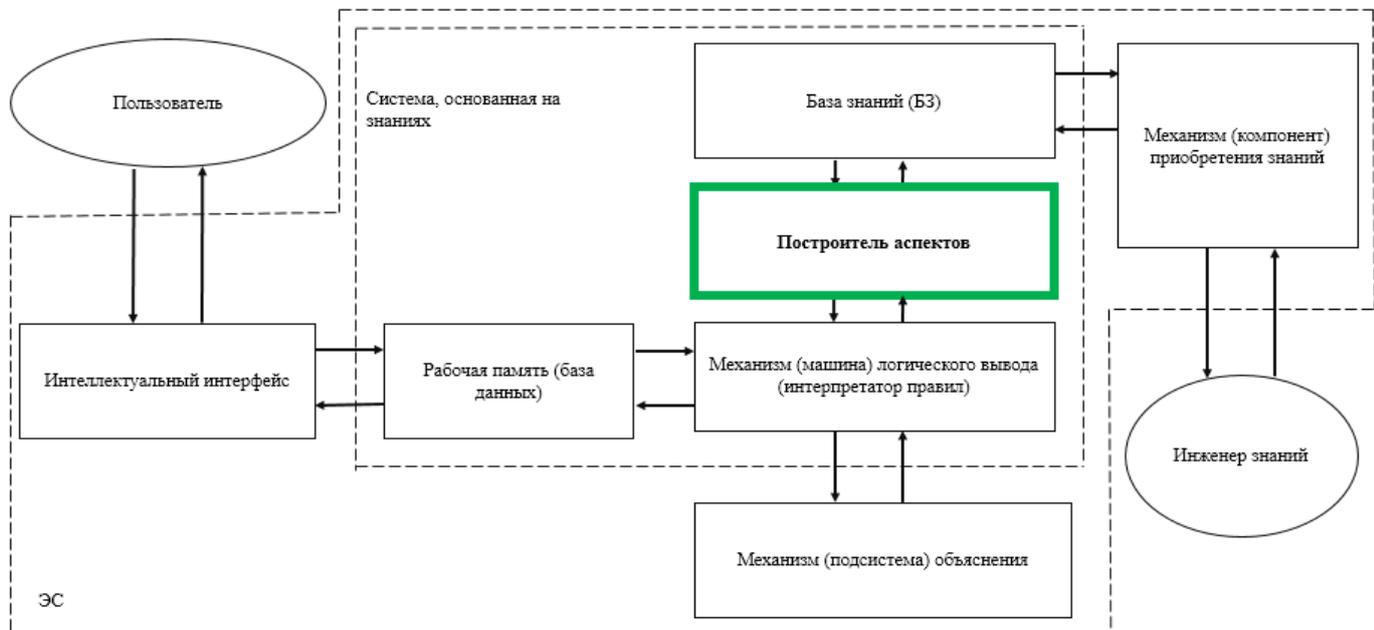


Рисунок 18. Модель экспертной ИС с применением АО-подхода

В модель экспертной ИС добавляется новый элемент – построитель аспектов. Построитель аспектов представляет собой подсистему интерпретации продукционных правил экспертной ИС с применением АО-подхода.

Построитель аспектов преобразует исходный набор продукционных правил в новый набор с использованием одного из алгоритмов извлечения аспектов из последовательности фактов. Выбор подходящего алгоритма осуществляется с использованием МАИ.

С учётом требования, касающегося отсутствия переработки исходной структуры БЗ и БД экспертной ИС, построитель аспектов:

- загружает из БД исходный набор продукционных правил;
- в интерпретатор правил передаёт набор правил, реорганизованных при помощи одного из алгоритмов извлечения аспектов;

- сохраняет в БД исходный набор продукционных правил, сделав обратное преобразование, реорганизованных при помощи одного из алгоритмов извлечения аспектов, продукционных правил.

Описание прототипа подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода приведено ниже.

#### 4.2. Описание реализации прототипа подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода

Прототип программной реализации экспертной ИС интеллектуальной поддержки специалистов испытательных лабораторий с применением аспектно-ориентированного подхода при сертификации средств защиты информации промышленных ИС базируется на технологии Spring Boot. Архитектура программы представлена на рисунке 19.

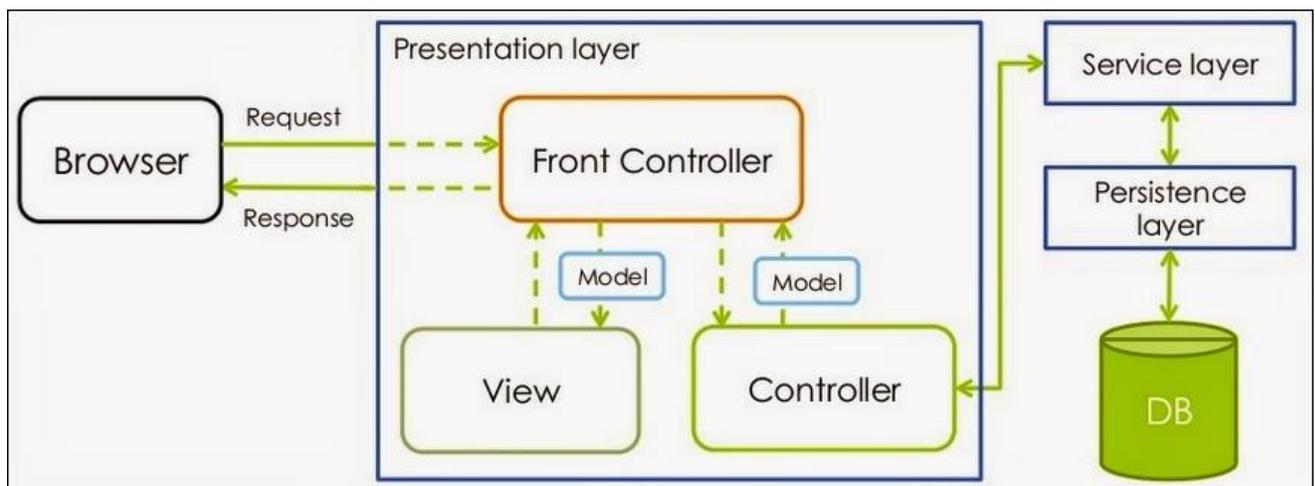


Рисунок 19. Архитектура программы

Экспертная система представлена блоком Service layer. Взаимодействие с пользователем осуществляется через браузер.

Подсистема интерпретации продукционных правил (построитель аспектов) экспертной ИС реализована на языке Java (Рисунок 20).

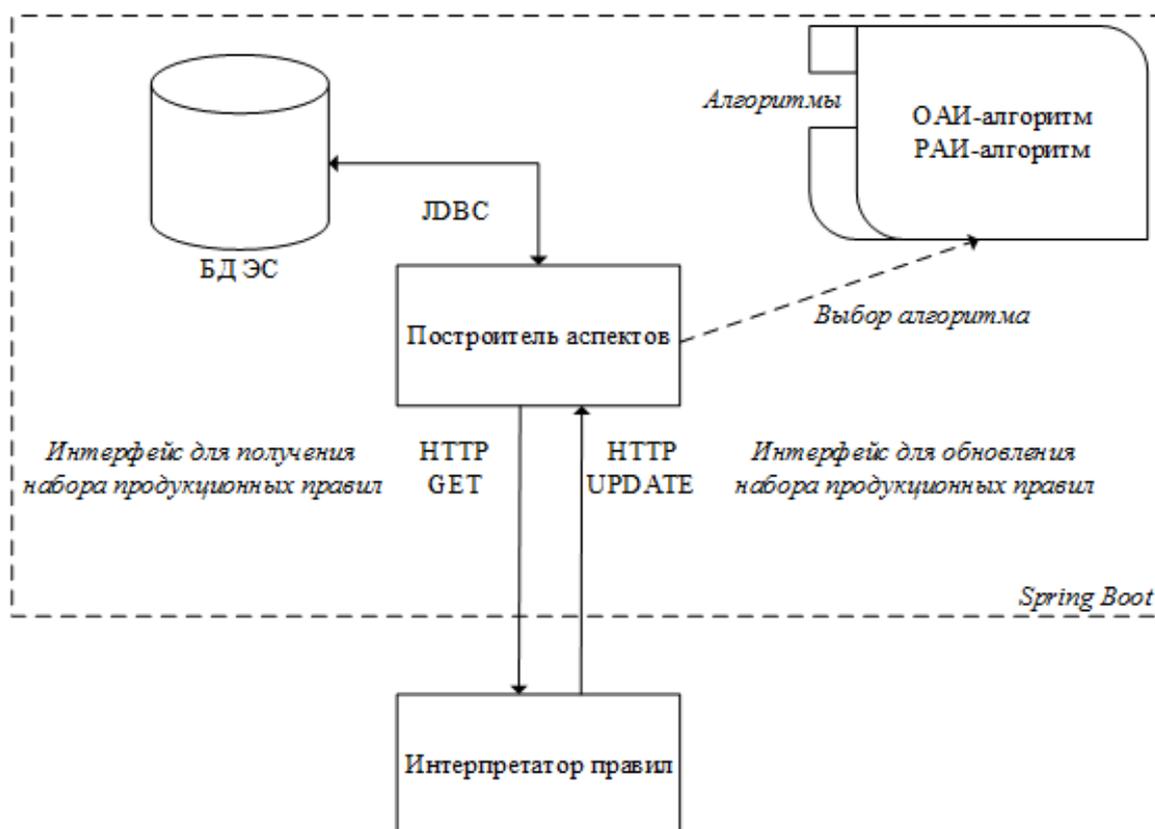


Рисунок 20. Подсистема интерпретации продукционных правил

Построитель аспектов выполняет подключение к БД экспертной ИС с помощью JDBC-коннектора (Java Database Connectivity) и загружает исходный набор продукционных правил. Для получения набора правил, реорганизованного при помощи одного из алгоритмов извлечения аспектов, интерпретатор правил с помощью HTTP метода GET в соответствии с REST-подходом (Representational State Transfer). Для сохранения состояния набора правил, реорганизованного при помощи одного из алгоритмов извлечения аспектов, набор правил преобразуется в исходную форму и сохраняется в БД экспертной ИС с помощью HTTP метода UPDATE.

Блок-схема взаимодействия классов реализации подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов, построенная с помощью встроенных средств среды разработки IntelliJ IDEA, представлена в Приложении Б.

Исходный код подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов

извлечения аспектов из последовательности фактов на языке Java представлен в Приложении В.

В ходе разработки использовались следующие библиотеки:

- Apache Commons IO (версия 2.6);
- Apache Commons Lang (версия 3.9);
- Google Guava (версия 27.1-jre);
- Jackson (версия 2.9.8);
- JUnit (версия 4.10);
- Hamcrest (версия 1.1).

Сборка проекта выполнялась с использованием фреймворка Apache Maven (версия 3.3.9).

#### **4.3. Выводы по главе**

В данной главе был представлен прототип подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода.

Были проанализированы необходимые изменения модели экспертной ИС в части интерпретации продукционных правил, базирующееся на применении АО-подхода с учётом требований к сертификации встраиваемого ПО АСУ ТП.

Был представлен исходный код подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов на языке Java и описание используемых библиотек и фреймворков.

## ЗАКЛЮЧЕНИЕ

Развитие промышленных предприятий в настоящее время невозможно без применения автоматизированных систем управления производственными и технологическими процессами (АСУ ТП). Наибольший экономический успех сегодня сопутствует только тем предприятиям, которые активно используют современные информационные технологии [57].

Встраиваемое программное обеспечение (ПО) АСУ ТП, помимо выполнения целевых функций, должно противостоять следующим угрозам обрабатываемой информации:

- неправомерный доступ
- уничтожение
- модифицирование
- блокирование
- копирование
- предоставление
- распространение
- иные неправомерные действия, в том числе деструктивные информационные воздействия (компьютерные атаки), следствием которых может стать нарушение функционирования АСУ ТП [63].

Авторы и исследователи Ю.А. Кудеяров, А.В. Шестаков, А.А. Дудыкин, А.Н. Паньков и др. выделяют следующие характерные особенности встраиваемого ПО АСУ ТП:

- значительный объем обрабатываемых информационных потоков, сложность и многогранность процессов обработки и передачи информации;
- многообразие функциональных возможностей;
- критическая важность бесперебойного поддержания работоспособности;

- болезненная, с точки зрения совместимости и технической поддержки, реакция на любые изменения в аппаратную или программную составляющие;
- необходимость особенного внимания к обеспечению триады свойств безопасности информационных ресурсов – конфиденциальности, целостности и доступности.

Аккумуляирование эмпирических знаний о принимаемых при управлении технологическими процессами решениях привело к использованию в составе встраиваемого ПО АСУ ТП производственных экспертных информационных систем (ИС) промышленного назначения [96]. Производственные экспертные ИС также называются системами на основе правил. Производственная модель чаще всего применяется в промышленных экспертных ИС ([47], [59]).

В силу представленных выше характерных особенностей встраиваемого ПО АСУ ТП, согласно [63], возникает необходимость осуществления контроля, которая находит своё отражение в процедуре сертификации промышленных ИС.

Сертификация средств защиты информации встраиваемого ПО АСУ ТП производится в соответствии с утвержденным приказом № 55 ФСТЭК России от 3 апреля 2018 г. «Положением о системе сертификации средств защиты информации».

При сертификации встраиваемого ПО АСУ ТП на первый план выступают сроки проведения всех необходимых проверок. Важность сосредоточения на уменьшении продолжительности сертификационных испытаний обусловлена невозможностью снижения конкурентоспособности предприятия.

Промышленность России представлена множеством отраслей и предприятий. Сложившееся обвальное нарастание массы самой разнообразной информации о технологиях промышленности, знаний и метазнаний о промышленных процессах наделяет современные экспертные ИС встраиваемого ПО АСУ ТП большим количеством производственных правил.

Авторы Валетов В.А., Орлова А.А., Третьяков С.Д. и др. рассматривали проблемы экспертных ИС и в качестве одного из актуальных недостатков

продукционных экспертных ИС выделяют меньшую эффективность процесса логического вывода, по сравнению с другими моделями представления знаний. Причину этого можно найти в том, что при логическом выводе требуется значительное количество времени на перебор и непроизводительные проверки применимости правил [8].

Представленные требования к встраиваемому ПО АСУ ТП показывают важность решения и актуальность проблемы повышения эффективности процесса логического вывода в продукционных экспертных ИС.

Цель диссертационной работы заключалась в поиске методов повышения эффективности процесса логического вывода в продукционных экспертных ИС при сертификации встраиваемого ПО АСУ ТП.

Исходя из цели, в диссертации были сформулированы и решены следующие задачи исследования:

1. Формирование особенностей сертификации продукционных экспертных ИС встраиваемого ПО АСУ ТП.
2. Анализ применяемых в настоящее время методов повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП.
3. Разработка алгоритмов решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП с использованием аспектно-ориентированного (АО) подхода.
4. Использование метода анализа иерархий (МАИ) для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в базе знаний (БЗ) экспертной ИС.
5. Описание изменений модели экспертной ИС в части интерпретации продукционных правил, базирующихся на применении АО-подхода, при сертификации встраиваемого ПО АСУ ТП.

6. Разработка прототипа подсистемы интерпретации производственных правил экспертной ИС с применением АО-подхода.

**В первой главе** были описаны особенности встраиваемого ПО АСУ ТП и промышленных производственных экспертных ИС.

Была описана структура типовой АСУ ТП.

Дано описание встраиваемому ПО и современным ИС, выделены характерные особенности встраиваемого ПО АСУ ТП, приведены особенности разработки встраиваемого ПО АСУ ТП.

Было представлено описание, основные особенности и причины возникновения экспертных ИС, приведён список наиболее известных экспертных ИС.

В данной главе также было дано описание процесса сертификации и были сформулированы следующие требования (особенности) предъявляемые к сертификации производственных экспертных ИС встраиваемого ПО АСУ ТП:

- уменьшение продолжительности сертификационных испытаний;
- необходимость в неизменности параметров технологического процесса и пределов значений параметров выполнения целевых функций, заложенных в БЗ производственной экспертной ИС встраиваемого ПО АСУ ТП.

Были описаны особенности логического вывода производственных экспертных ИС промышленного назначения и их главные недостатки, среди которых:

- значительное количество времени на перебор и непроизводительные проверки применимости правил (продукций);
- сложность понимания взаимосвязей объектов;
- продолжительность вычислений и времени логического вывода увеличивается вместе с количеством числа правил и число записей, находящихся в базе фактов;
- недетерминированность (неоднозначность);

- при большом количестве правил становится сложно отслеживать непротиворечивость базы знаний.

Исходя из этого, была поставлена задача исследования, заключающаяся в нахождении методов повышения эффективности процесса логического вывода в производственных экспертных ИС при сертификации встраиваемого ПО АСУ ТП.

**Во второй главе** были представлены общеизвестные методы решения задачи повышения скорости вычислительных операций в производственных экспертных ИС встраиваемого ПО АСУ ТП:

- фильтры;
- фреймы;
- дуальные экспертные системы;
- нейронные сети;
- генетические алгоритмы;
- Rete-алгоритм.

Было дано объяснение невозможности применимости каждого из данных методов при сертификации производственных экспертных ИС встраиваемого ПО АСУ ТП: фильтры, фреймы, дуальные экспертные системы, нейронные сети и генетические алгоритмы не могут быть применены поскольку среди предъявленных требований к сертификации экспертных ИС встраиваемого ПО АСУ ТП важна неизменность параметров технологического процесса и пределов значений параметров выполнения целевых функций, заложенных в БЗ производственной экспертной ИС встраиваемого ПО АСУ ТП; Rete-алгоритм не применим в силу большой потребности в оперативной памяти.

Были представлены цель, особенности и предназначение аспектно-ориентированного подхода. Дано описание преимуществ и недостатков АОП, приведены фундаментальные понятия АОП.

В данной главе также были приведены возможные альтернативы АОП – паттерны проектирования и DI/IoC-фреймворки.

Было представлено описание разработанных алгоритмов решения задачи повышения скорости вычислительных операций в производственных экспертных ИС встраиваемого ПО АСУ ТП, базирующихся на применении АО-подхода:

- одноитерационный алгоритм извлечения аспектов из последовательности фактов;
- рекурсивный алгоритм извлечения аспектов из последовательности фактов.

**Третья глава** была посвящена анализу критериев повышения скорости вычислительных операций в производственных экспертных ИС.

Были представлены исходные данные. В качестве исходных данных выбрана производственная экспертная ИС выбора мер защиты информации в АСУ ТП, предъявляемых для соответствующего класса защищенности АСУ ТП в соответствии с требованиями приказа Федеральной службы по техническому и экспортному контролю от 14 марта 2014 г. № 31 [63], применяемая для интеллектуальной поддержки специалистов испытательной лаборатории ЗАО ГИЦ ПС ВТ при сертификации встраиваемого ПО промышленных ИС.

Были представлены граф логического вывода набора производственных правил и значения скорости логического вывода в БЗ экспертной ИС для исходных данных и для данных, модифицированных с помощью разработанных алгоритмов решения задачи повышения скорости вычислительных операций в производственных экспертных ИС с использованием АО-подхода.

Приведено описание использования МАИ для обоснования выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества производственных правил в БЗ экспертной ИС согласно выделенным критериям повышения скорости вычислительных операций в производственных экспертных ИС: скорость формирования, занимаемый объем памяти, затраты времени.

**В четвертой главе** был представлен прототип подсистемы интерпретации производственных правил экспертной ИС с применением АО-подхода и

проанализированы изменения модели экспертной ИС в части интерпретации продукционных правил, базирующееся на применении АО-подхода.

Была предложена последовательность интерпретации продукционных правил.

Приведено описание изменений классической статической модели экспертной ИС в части интерпретации продукционных правил, базирующееся на применении АО-подхода.

В данной главе также была представлена блок-схема взаимодействия классов реализации подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов, построенная с помощью встроенных средств среды разработки IntelliJ IDEA и приведён исходный код подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов на языке Java.

В диссертации были получены следующие результаты, характеризующиеся научной новизной:

1. Предложенные одноитерационный и рекурсивный алгоритмы решения задачи повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП отличаются применением АО-подхода.
2. Обоснование выбора наиболее оптимального алгоритма извлечения аспектов из последовательности фактов в зависимости от количества продукционных правил в БЗ экспертной ИС встраиваемого ПО АСУ ТП отличается использованием МАИ.
3. Изменение модели экспертной ИС встраиваемого ПО АСУ ТП в части интерпретации продукционных правил отличается применением одноитерационного и рекурсивного алгоритмов решения задачи повышения скорости логического вывода в продукционных экспертных ИС.

4. Прототип подсистемы интерпретации продукционных правил экспертной ИС, применяемый при сертификации встраиваемого ПО АСУ ТП, отличается применением АО-подхода.

Рассчитанное значение глобального приоритета для каждой из альтернатив представлено в таблице 17.

Таблица 17. Определение наилучшей альтернативы

<i>Определение наилучшей альтернативы</i>	<i>Критерии</i>			<i>Глобальные приоритеты</i>
<i>Альтернативы</i>	Скорость формирования	Занимаемый объем памяти	Затраты времени	
	<i>Значение вектора приоритета</i>			
	0,570	0,333	0,097	
Без применения алгоритма	2,289	0,342	0,261	<b>1,443</b>
ОАИ-алгоритм	0,874	2,154	2,884	<b>1,496</b>
РАИ-алгоритм	0,500	1,357	1,326	<b>0,866</b>

Выбранной считается альтернатива с максимальным значением глобального приоритета. Таким образом, для БЗ экспертной ИС состоящей из 10 продукционных правил, необходимо выбрать ОАИ-алгоритм.

Повторив данные вычисления для БЗ экспертной ИС из 25, 50, 100, 250, 500 и 1000 продукционных правил, получим гистограмму, отображающую выбранные значения глобальных приоритетов, представленную на рисунке 21.

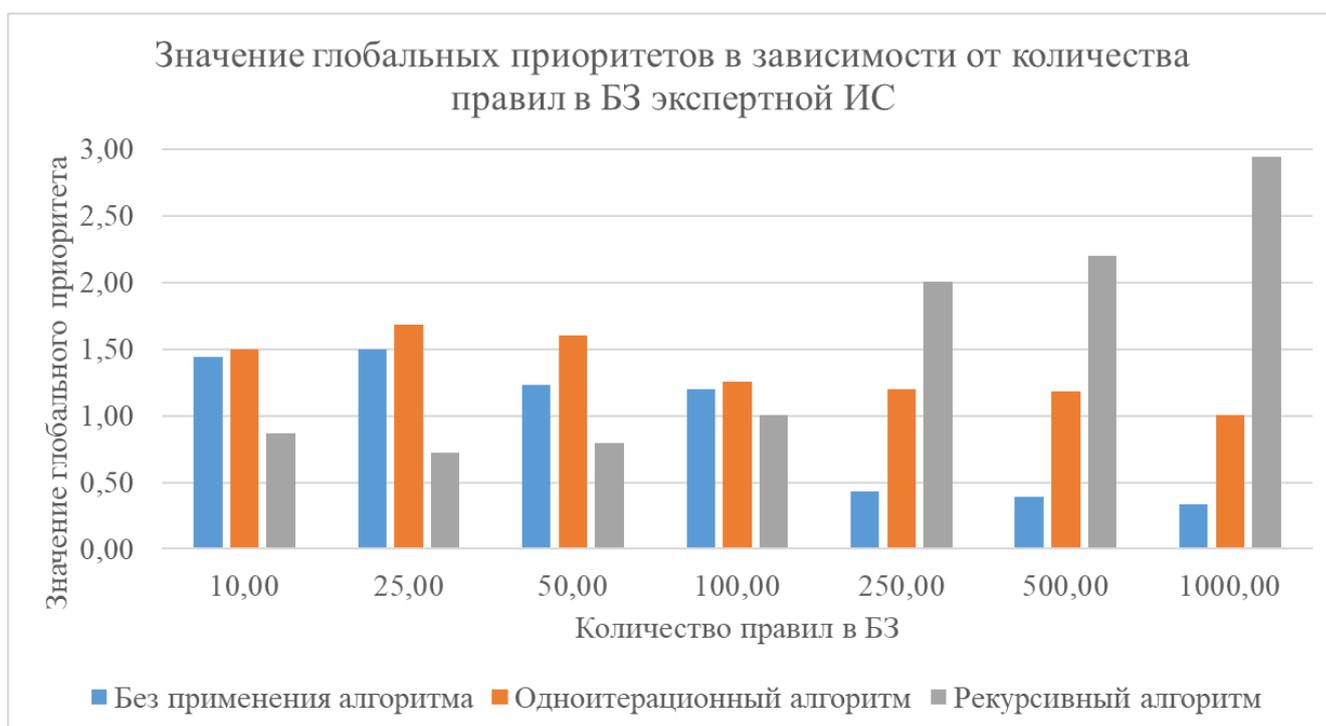


Рисунок 21. Гистограмма, отображающая выбранные значения глобальных приоритетов для БЗ экспертной ИС из 10, 25, 50, 100, 250, 500 и 1000 продукционных правил

Из данной гистограммы видно, что с увеличением количества продукционных правил в БЗ экспертной ИС до 500 правил, лучше всего начинает себя проявлять РАИ-алгоритм.

Результаты, полученные для общего случая, представлены в таблице 18.

Таблица 18. Результаты выбора алгоритма извлечения аспектов из последовательности фактов в зависимости от количества правил в БЗ экспертной ИС

№ п/п	Количество продукционных правил	Рекомендуемый алгоритм
1	10	ОАИ-алгоритм
2	25	ОАИ-алгоритм
3	50	ОАИ-алгоритм
4	100	ОАИ-алгоритм
5	250	ОАИ-алгоритм
6	500	РАИ-алгоритм
7	1000	РАИ-алгоритм

Скорость логического вывода в БЗ экспертной ИС после применения алгоритмов, разработанных с использованием АО-подхода увеличилась (Рисунок 22).

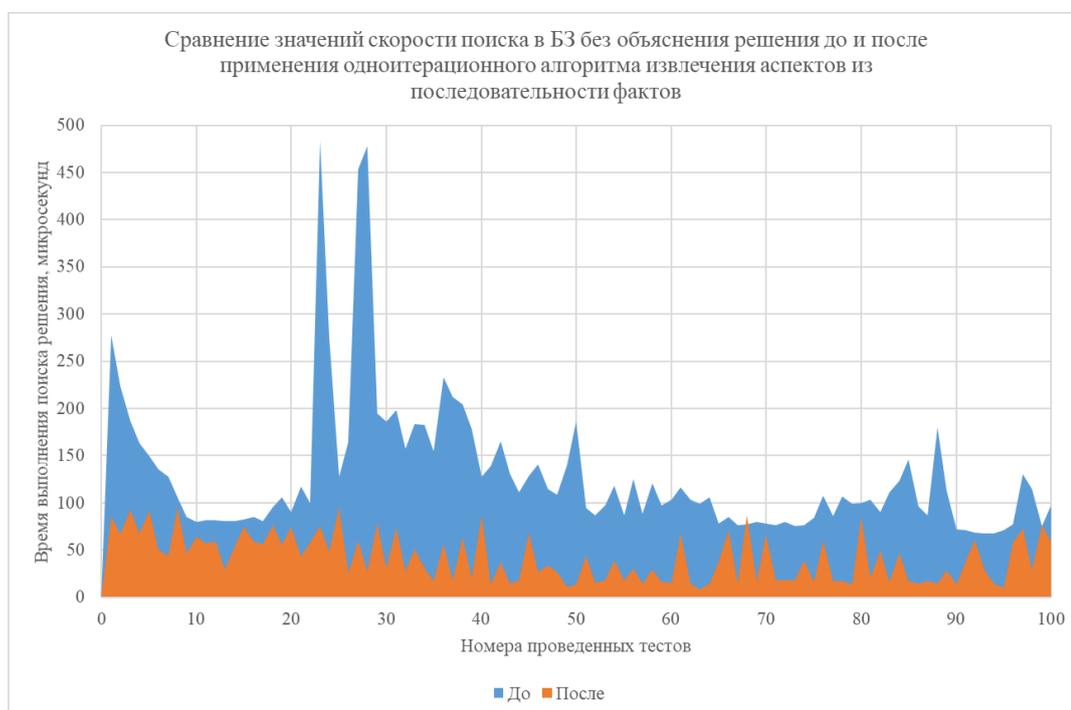


Рисунок 22. Сравнение значений скорости логического вывода в БЗ экспертной ИС из 322 правил до и после применения ОАИ-алгоритма

Выделение сквозных (пересекающихся) фактов в аспекты позволило избавиться от непроизводительной проверки применимости правил, сократив количество операций при поиске решения и избавившись от перебора всех фактов, что и привело к повышению скорости логического вывода.

Перспективой развития данного исследования является изучение возможности применения АО-подхода для других типов экспертных ИС, а также использование АО-подхода совместно с другими известными методами повышения скорости вычислительных операций в продукционных экспертных ИС встраиваемого ПО АСУ ТП:

- фильтры;
- фреймы;
- дуальные экспертные системы;
- нейронные сети;

- генетические алгоритмы;
- Rete-алгоритм.

Кроме того, рассматривается вариант использования разработанных алгоритмов в работе экспертных ИС систем обнаружения вторжений (IPS, Intrusion Prevention System). Сокращение количества операций при поиске решения и избавление от перебора всех фактов позволит увеличить скорость реакции на инциденты, что для систем обнаружения вторжений является важным.

## СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

<b>AOSD</b>	–	Aspect-oriented Software Development
<b>DI</b>	–	Dependency injection
<b>HTTP</b>	–	HyperText Transfer Protocol
<b>IDE</b>	–	Integrated Development Environment
<b>IoC</b>	–	Inversion of Control
<b>IPS</b>	–	Intrusion Prevention System
<b>JDBC</b>	–	Java Database Connectivity
<b>REST</b>	–	Representational State Transfer
<b>SCADA</b>	–	Supervisory Control And Data Acquisition
<b>АО</b>	–	Аспектно-ориентированный
<b>АОП</b>	–	Аспектно-ориентированное программирование
<b>АС</b>	–	Автоматизированная система
<b>АСУ ТП</b>	–	Автоматизированная система управления производственными и технологическими процессами
<b>БД</b>	–	База данных
<b>БЗ</b>	–	База знаний
<b>ВАК</b>	–	Высшая аттестационная комиссия
<b>ЗАО</b>	–	Закрытое акционерное общество
<b>ИС</b>	–	Информационная система
<b>МАИ</b>	–	Метод анализа иерархий
<b>МПЗ</b>	–	Модель представления знаний
<b>ПО</b>	–	Программное обеспечение
<b>СОВ</b>	–	систем обнаружения вторжений
<b>ФСТЭК</b>	–	Федеральная служба по техническому и экспортному контролю
<b>ЭВМ</b>	–	Электронно-вычислительная машина

## СПИСОК ЛИТЕРАТУРЫ

1. Аксенов, С.В., Новосельцев, В.Б. Организация и использование нейронных сетей (методы и технологии) / Под общ. ред. В.Б. Новосельцева. – Томск: Изд-во НТЛ, 2006. – 128 с.
2. Антонов, А.В. Системный анализ : учебник для вузов / А.В. Антонов. – М.: Высш. шк., 2004. – 454 с.
3. Барский, А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. – 176 с.
4. Беляков, И.А. Метод поддержки принятия решения о безопасности программного обеспечения // Дис. канд. техн. наук: 05.13.19 / Беляков И.А. – СПб: Петербургский государственный университет путей сообщения, 2013. – 197 с.
5. Беркинблит, М.Б. Нейронные сети. – М.: МИРОС и ВЗМШ РАО, 1993. – 96 с.
6. Большая советская [Электронный ресурс]. – Режим доступа: <http://www.rubricon.com>. – (Дата обращения: 22.04.2019).
7. Братко, И.М. Алгоритмы искусственного интеллекта на языке PROLOG / И.М. Братко. – М.: Вильямс, 2001. – 640 с.
8. Валетов, В.А., Орлова, А.А., Третьяков, С.Д. Интеллектуальные технологии производства приборов и систем : учебное пособие. – СПб: СПб ГУИТМО, 2008. – 134 с.
9. Волкова, В.Н., Денисов, А.А. Теория систем и системный анализ : учебник для академического бакалавриата / В. Н. Волкова, А. А. Денисов. – 2-е изд., перераб. и доп. – М.: Издательство Юрайт, 2014. – 616 с.
10. Воронов А.А. Основы теории автоматического управления. Часть 1. Линейные системы регулирования одной величины. – М.-Ленинград: Изд-во «Энергия», 1965. – 396 с.
11. Гаврилова, Т.А., Хорошевский, В.Ф. Базы знаний интеллектуальных систем : учебник. – СПб.: Питер, 2000. – 384 с.

12. Гаврилова, Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. – М.: Радио и связь, 1992. – 200 с.

13. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – СПб.: Питер, 2007. – 366 с.

14. Герасимова, Е.Б., Герасимов, Б.И. Метрология, стандартизация и сертификация: учебное пособие / Е.Б. Герасимова, Б.И. Герасимов. – М.: ФОРУМ, 2010. – 224 с.

15. Гладков, Л.А., Курейчик, В.В., Курейчик, В.М. Генетические алгоритмы : учебное пособие. – 2-е изд. – М: Физматлит, 2006. – 320 с.

16. Головчинер, М.Н., Введение в системы знаний : курс лекций. – Томск: ТГУ, 2015. – 97 с.

17. Гончаров, А.А. Аспектно-ориентированное программирование в контексте решения вопросов повышения эффективности экономических показателей IT-проектов / А.А. Гончаров, Н.А. Семенов // Программные продукты и системы. - 2016. - № 3.

18. Гончаров, А.А. Аспектно-ориентированный подход как метод повышения скорости логического вывода продукционных экспертных систем / А.А. Гончаров // Международный конгресс по интеллектуальным системам и информационным технологиям («IS&IT'18»). – 2018. – Т. 1.

19. Гончаров, А.А. Использование нового стандарта ГОСТ Р 56939–2016 как основной шаг к безрисковой сертификации средств защиты информации / А.А. Гончаров // Материалы научно-практической конференции «Теоретические исследования и экспериментальные разработки студентов и аспирантов ТвГТУ», приуроченной ко Дню российской науки. – 2017. – Т. 3.

20. Гончаров, А.А. Использование технологий виртуализации на промышленных предприятиях как основа перехода к гибридным облачным вычислениям / А.А. Гончаров // Материалы научно-практической конференции

«Теоретические исследования и экспериментальные разработки студентов и аспирантов ТвГТУ», приуроченной ко Дню российской науки. – 2015. – Т. 3.

21. Гончаров, А.А. Модель доступа к информационным ресурсам предприятия с помощью подсистемы Oracle Entitlements Server на примере модуля безопасности IBM WebSphere / А.А. Гончаров // Сборник научных трудов XII-ой Международной научно-практической конференции «Логистика и экономика ресурсосбережения и энергосбережения в промышленности» (МНПК «ЛЭРЭП-9-2015»). – 2015. – Т. 3.

22. Гончаров, А.А. Повышение скорости логического вывода продукционных экспертных систем путём использования аспектно-ориентированного подхода / А.А. Гончаров, Н.А. Семенов // Программные продукты и системы. – 2018. – № 3.

23. Гончаров, А.А. Подход к приобретению базовых знаний для создания экспертной системы проведения инспекционного контроля / А.А. Гончаров // Российская конференция с международным участием «Искусственный интеллект, когнитивное моделирование и интеллектуальная робототехника» (ИИКМИР'2019). – 2019.

24. Гончаров, А.А. Применение аспектно-ориентированного подхода в экспертных системах промышленного назначения / А.А. Гончаров, Н.А. Семенов // IX-ая Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (Коломна, 19-22 мая 2019 г.).

25. Гончаров, А.А., Семенов, Н.А. Повышение скорости логического вывода в экспертной системе / А.А. Гончаров, Н.А. Семенов // Научный журнал «Мягкие измерения и вычисления». – 2018. – №. 8.

26. Горожанина, Е.И. Нейронные сети : учебное пособие / Е.И. Горожанина. – Самара. ФГБОУ ВО ПГУТИ, 2017. – 84 с.

27. ГОСТ Р 50922-2006 «Защита информации. Основные термины и определения»

28. Данелян, Т.Я. Теория систем и системный анализ : учебно-методический комплекс / Т.Я. Данелян. – М.: Изд. центр ЕАОИ, 2010. – 303 с.
29. Джарратано, Д., Райли Г. Экспертные системы. Принципы разработки и программирование // 4-е издание. – М.: ООО «И.Д. Вильямс», 2007. – 1152 с.
30. Джексон, П. Введение в экспертные системы / П. Джексон. – 3-е изд. – М.: Вильямс. – 393 с.
31. Диксон, Д. Проектирование систем: изобретательство, анализ и принятие решений. – М.: Изд-во «Мир», 1969. – 442 с.
32. Димов, Ю.В. Метрология, стандартизация и сертификация. – СПб.: Питер, 2010. – 464 с.
33. Домарев, В.В. Безопасность информационных технологий. Методология создания систем защиты [Электронный ресурс]. – Режим доступа: <http://w15408.narod.ru/books/Domarev/domarev.htm>. – (Дата обращения: 15.03.2018).
34. Донской, В.И. Дуальные экспертные системы // Изв. РАН. Техническая кибернетика. – 1993. – №5. – С. 111-119.
35. Жилин, Д. Теория систем. Опыт построения курса. – М.: URSS: ЛЕНАНД, 2017. – 176 с.
36. Зайцев, Н.Г. Информационное и математическое обеспечение АСУП. – К.: "Техника", 1974. – 144 с.
37. Золотов, С.И. Интеллектуальные информационные системы : учебное пособие для студентов вузов / С.И. Золотов. – Воронеж: Научная книга, 2008. – 140 с.
38. Ильин, В.В. Управление эффективностью внедрения информационных систем. – М.: Lennex Corp, 2014. – 289 с.
39. Йенер, М., Фидом, А. Java EE. Паттерны проектирования для профессионалов. – СПб.: «Питер», 2016. – С. 122-123.
40. Каллан, Р. Основные концепции нейронных сетей / Роберт Каллан. – М.: Вильямс, 2003. – 287 с.

41. Капустин, Н.М. Автоматизация производственных процессов в машиностроении. – М.: Высшая школа, 2004. – 415 с.
42. Капустин, Н.М., Кузнецов, П.М., Схиртладзе, А.Г. Автоматизация производственных процессов в машиностроении : учеб. для втузов / Н.М. Капустин, П.М. Кузнецов, А.Г. Схиртладзе и др.; Под ред. Н.М. Капустина. – М.: Высш. шк., 2004. – 415 с.
43. Катасёв, А.С. Нейронечеткая модель формирования баз знаний экспертных систем с генетическим алгоритмом обучения / А.С. Катасёв, Ч.Ф. Ахатова // Проблемы управления и моделирования в сложных системах: Труды XII Междунар. конф. – Самара, 2010. – С. 615-621.
44. Каткова, А.Л. Экспертные системы : учебно-методическое пособие / А.Л. Каткова // Шадр. гос. пед. ин-т. – Шадринск, 2011. – 92 с.
45. Кондрашов, К.С., Жураковский, В.Н., Силин, С.И. Принципы проектирования встраиваемых систем на основе программных средств // 2307-0595, Инженерный вестник, Ноябрь, №11, 2014. – 9 с.
46. Котов, С.Л., Палюх, Б.В., Федченко, С.Л. Разработка, стандартизация и сертификация программных средств и информационных технологий и систем : учебное пособие. 1-е изд. – Тверь: ТГТУ, 2006. – 104 с.
47. Кузнецов, С.М. Информационные технологии : учебное пособие / С.М. Кузнецов. – Новосибирск: НГТУ, 2011. – 144 с.
48. Лазарева, Т.Я., Мартемьянов, Ю.Ф. Основы теории автоматического управления : учебное пособие. 2-е изд., перераб. и доп. – Тамбов: Изд-во Тамб. гос. техн. ун-та, 2004. – 352 с.
49. Лебедев, К.Н. Автоматизированные системы управления технологическими процессами : учебное пособие / К.Н. Лебедев. – Зерноград, ФГОУ ВПО АЧГАА, 2008. – 117 с.
50. Липаев, В.В. Проектирование и производство сложных заказных программных продуктов. – М.: СИНТЕГ, 2011. – 408 с.

51. Лорьер, Ж.-Л. Системы искусственного интеллекта. – М.: Изд-во «Мир», 1991. – 568 с.
52. Люгер, Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание. – М.: Вильямс, 2003. – 864 с.
53. Макаренко, С.И. Интеллектуальные информационные системы : учебное пособие. – Ставрополь: СФ МГГУ им. М. А. Шолохова, 2009. – 206 с.
54. Малыхина, М.П., Бегман, Ю.В. Нейросетевая экспертная система на основе прецедентов для решения проблем обслуживания абонентов сотовой сети // Известия вузов. Северо-Кавказский регион. Серия: Технические науки. 2009. №3 [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/neyrosetevaya-ekspertnaya-sistema-na-osnove-pretседentov-dlya-resheniya-problem-obsluzhivaniya-abonentov-sotovoy-seti>. – (Дата обращения: 22.04.2019).
55. Марков, А., Никулин, М., Цирлов, В. Сертификация средств защиты персональных данных: революция или эволюция? // Защита информации. INSIDE. – 2008. – № 5. – 7 с.
56. Марков, А.С., Цирлов, В.Л., Барабанов, А.В. Методы оценки несоответствия средств защиты информации / А.С. Марков, В.Л. Цирлов, А.В. Барабанов; под ред. А.С. Маркова. – М.: Радио и связь, 2012. – 192 с.
57. Матяш, С.А. Информационные технологии управления : курс лекций / С.А. Матяш. – М.-Берлин: Директ-Медиа, 2014. – 537 с.
58. Мигас, С.С. Интеллектуальные информационные системы : конспект лекций / С.С. Мигас. – СПб: СПбГИЭУ, 2009. – 160 с.
59. Морозова, В.А., Паутов, В.И. Представление знаний в экспертных системах : учебное пособие / В.А. Морозова, В.И. Паутов. – Екатеринбург: Изд-во Урал. ун-та, 2017. – 120 с.
60. Муромцев, Д.И. Введение в технологию экспертных систем. – СПб: СПб ГУ ИТМО, 2005. – 93 с.

61. Николаев, В.И., Брук, В.М. Системотехника: методы и приложения. – Л.: Машиностроение, Ленингр. отд-ние, 1985. – 199 с.

62. Нильсон, Н. Искусственный интеллект. Методы поиска решений. М.: Мир, 1973. – 273 с.

63. Об утверждении требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды [Электронный ресурс]: приказ Федеральной службы по техническому и экспортному контролю от 14 марта 2014 г. № 31 (в ред. Приказа ФСТЭК России от 23.03.2017 № 49). – Режим доступа: <https://fstec.ru/component/attachments/download/714>. – (Дата обращения: 28.04.2019).

64. Остроух, А.В. Интеллектуальные системы / А.В. Остроух. – Красноярск: Научно-инновационный центр, 2015. – 110 с.

65. Панченко, Т.В. Генетические алгоритмы : учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.

66. Положение о системе сертификации средств защиты информации (утверждено приказом ФСТЭК России от 3 апреля 2018 г. № 55) [Электронный ресурс]. – Режим доступа: <https://fstec.ru/component/attachments/download/1883>. – (Дата обращения: 22.04.2019).

67. Поляков, С.Д. Сертификация программной продукции. Методология оценки соответствия. Монография / ГОУ ВПО Моск. гос. техн. ун-т «Станкин», ф-т ИНТЕХ, каф. «Информационные системы». – М.: МГТУ «Станкин», 2011. – 295 с.

68. Попов, Э.В., Фоминых, И.Б., Кисель, Е.Б. Статические и динамические экспертные системы : учебное пособие / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. – М.: Финансы и статистика, 1996. – 320 с.

69. Примеры экспертных систем [Электронный ресурс]. – Режим доступа: <http://www.aiportal.ru/articles/expert-systems/examples-expsys.html>. – (Дата обращения: 22.04.2019).

70. Рутковская, Д., Пилиньский, М., Рутковский, Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия - Телеком, 2006. – 452 с.

71. Саати, Т. Принятие решений. Метод анализа иерархий / Перевод с английского Р. Г. Вачнадзе. – Москва «Радио и связь», 1993. – 278 с.

72. Сидельников, Ю.В. Системный анализ экспертного прогнозирования. Монография. – М.: МАИ, 2007. – 453 с.

73. Сидоркевич, М.В. Применение методов аспектно-ориентированного программирования для реализации паттернов объектно-ориентированного программирования // Современные технологии в задачах управления, автоматизации и обработки информации: Труды XIX международного научно-технического семинара. Алушта, 2010. – С. 213-214.

74. Федеральный закон «О техническом регулировании» от 27.12.2002 N 184-ФЗ (в ред. Федеральных законов от 09.05.2005 N 45-ФЗ, от 01.05.2007 N 65-ФЗ, от 01.12.2007 N 309-ФЗ, от 23.07.2008 N 160-ФЗ, от 18.07.2009 N 189-ФЗ, от 23.11.2009 N 261-ФЗ, от 30.12.2009 N 384-ФЗ, от 30.12.2009 N 385-ФЗ, от 28.09.2010 N 243-ФЗ, от 21.07.2011 N 255-ФЗ, от 30.11.2011 N 347-ФЗ, от 06.12.2011 N 409-ФЗ, от 28.07.2012 N 133-ФЗ, от 03.12.2012 N 236-ФЗ, от 02.07.2013 N 185-ФЗ, от 23.07.2013 N 238-ФЗ, от 28.12.2013 N 396-ФЗ, от 23.06.2014 N 160-ФЗ, от 20.04.2015 N 102-ФЗ, от 29.06.2015 N 160-ФЗ, от 13.07.2015 N 216-ФЗ, от 28.11.2015 N 358-ФЗ, от 05.04.2016 N 104-ФЗ, от 01.07.2017 N 141-ФЗ, от 29.07.2017 N 216-ФЗ)

75. Частиков, А.П. Разработка экспертных систем. Среда CLIPS / А.П. Частиков, Т.А. Гаврилова, Д.Л. Белов. – СПб: БХВ-Петербург, 2003. – 608 с.

76. Шаньгин, В.Ф. Информационная безопасность компьютерных систем и сетей : учебное пособие. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2014. – 416 с.

77. Экспертная система [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Экспертная\\_система](https://ru.wikipedia.org/wiki/Экспертная_система). – (Дата обращения: 22.04.2019).

78. Юрин, А.М. Экспертные системы / А.М. Юрин. – Казань: Казан. ун-т, 2015. – 19 с.

79. Artificial Intelligence: The Technology of Expert Systems [Электронный ресурс]. – Режим доступа: <https://pubs.acs.org/doi/pdfplus/10.1021/bk-1986-0306.ch001?src=recsys>. – (Дата обращения: 22.04.2019).

80. Building Knowledge Automation Expert Systems with Exsys Corvid [Электронный ресурс]. – Режим доступа: <http://www.exsys.com/pdf/BookDetails.pdf>. – (Дата обращения: 22.04.2019).

81. Dijkstra, E.W. Selected Writings on Computing: A Personal Perspective. N.Y.: Springer Verlag, 1982. – P. 60 – 66.

82. Doorenbos, R.B. Production Matching for Large Learning Systems [Электронный ресурс]. – Режим доступа: <http://reports-archive.adm.cs.cmu.edu/anon/1995/CMU-CS-95-113.pdf>. – (Дата обращения: 22.04.2019).

83. Elrad, T., Aksit, M.M., Kiczales, G., et al. Discussing Aspects of AOP // Communications of ACM. 2001. October. V. 44. No. 10. – P. 33 – 38

84. Elrad, T., Filman, R.E., Bader, A. Aspect-Oriented Programming // Communications of ACM. 2001. October. V. 44. No. 10. – P. 29 – 32.

85. Gradecki, J.D., Lesiecki, N. Mastering AspectJ: Aspect-Oriented Programming in Java, Wiley Publ., 2003. – 456 p.

86. Inversion of Control with the Managed Extensibility Framework (MEF) [Электронный ресурс]. – Режим доступа: <http://www.informit.com/articles/article.aspx?p=1635818&seqNum=2>. – (Дата обращения: 22.04.2019).

87. Kiczales, G., Lamping, J., Mendhekar, A. et al. Aspect-oriented programming. Xerox PARC, 1997. – P. 25.

88. Laddad, R., Johnson, R. AspectJ in Action: Enterprise AOP with Spring Applications, Manning Publ., 2009. – 568 p.

89. Lau, P. An Architectural Framework for Construction of A Flexible Expert System for Engineering Selection [Электронный ресурс]. – Режим доступа: [https://www.collectionscanada.gc.ca/obj/s4/f2/dsk2/tape16/PQDD\\_0005/MQ30500.pdf](https://www.collectionscanada.gc.ca/obj/s4/f2/dsk2/tape16/PQDD_0005/MQ30500.pdf). – (Дата обращения: 11.02.2018).

90. Lucas, P. Principles of Expert Systems [Электронный ресурс]. – Режим доступа: <https://www.cs.ru.nl/P.Lucas/proe.pdf>. – (Дата обращения: 10.02.2019).

91. McDermott, D. Planning and Acting. Cognitive Science [Электронный ресурс]. – Режим доступа: <http://www.jimdavies.org/summaries/mcdermott1978-2.html>. – (Дата обращения: 22.04.2019).

92. Merritt, D. Building Expert Systems in Prolog [Электронный ресурс]. – Режим доступа: [http://www.amzi.com/distribution/files/xsip\\_book.pdf](http://www.amzi.com/distribution/files/xsip_book.pdf). – (Дата обращения: 20.02.2019).

93. Minsky, M. A Framework for Representing Knowledge [Электронный ресурс]. – Режим доступа: <https://courses.media.mit.edu/2004spring/mas966/Minsky%201974%20Framework%20for%20knowledge.pdf>. – (Дата обращения: 22.04.2019).

94. Pawlak, R. Foundations of AOP for J2EE Development / Renaud Pawlak, Lionel Seinturier, Jean-Philippe Retaille. – USA: Apress, 2005. – P. 328.

95. Positive Technologies. Количество уязвимостей в АСУ ТП во всем мире выросло на 30% [Электронный ресурс]. – Режим доступа: <https://www.ptsecurity.com/ru-ru/about/news/kolichestvo-uyazvimostej-v-asu-tp-vo-vsem-mire-vyroslo-na-30-procentov/>. – (Дата обращения: 22.04.2019).

96. Sasikumar M., Ramani S., Muthu Raman S., Chandrasekar R. A Practical Introduction to Rule Based Expert Systems [Электронный ресурс]. – ResearchGate, 2007. – Режим доступа: [https://www.researchgate.net/profile/Srinivasan\\_Ramani/publication/265038834\\_A\\_Practical\\_Introduction\\_to\\_Rule\\_Based\\_](https://www.researchgate.net/profile/Srinivasan_Ramani/publication/265038834_A_Practical_Introduction_to_Rule_Based_)

Expert\_Systems/links/551faa1d0cf2a2d9e1407dce/A-Practical-Introduction-to-Rule-Based-Expert-Systems.pdf. – (Дата обращения: 20.03.2019).

97. Tarditi, D., Morrisett, G., Cheng, P., Stone, C., Harper, R., Lee, P. A Type-Directed Optimizing Compiler for ML [Электронный ресурс]. – Режим доступа: <https://www.cs.cmu.edu/~rwh/papers/til/tr.pdf>. – (Дата обращения: 22.04.2019).

98. The Don't Repeat Yourself Principle and the Wormhole Anti-Pattern [Электронный ресурс]. – Режим доступа: <http://codebetter.com/jeremymiller/2007/03/22/the-dont-repeat-yourself-principle-and-the-wormhole-anti-pattern/>. – (Дата обращения: 05.06.2016).

99. Volpano, D., Smith, G. A Type-Based Approach to Program Security [Электронный ресурс]. – Режим доступа: <https://users.cs.fiu.edu/~smithg/papers/tapsoft97.pdf>. – (Дата обращения: 22.04.2019).

100. Yang, H. Software Reuse in the Emerging Cloud Computing Era. – Information Science Reference, 2012. – P. 54.

## ПРИЛОЖЕНИЕ А

Требования из [63] можно представить в виде объектов правил продукции (Таблица В.1) и объектов действий продукции (Таблица В.2) и получить исходный набор продукционных правил, представленный в листинге В.1:

```
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action1;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action2;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action3;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action4;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action5;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action6;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr1 ТО action8;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action9;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action10;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action11;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action13;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action14;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action15;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action19;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action20;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action22;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr2 ТО action23;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr4 ТО action28;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr4 ТО action29;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr4 ТО action30;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr4 ТО action33;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr4 ТО action35;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr4 ТО action36;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action37;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action38;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action39;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action40;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action41;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action43;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action44;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action45;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr5 ТО action47;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr6 ТО action49;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr6 ТО action50;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr6 ТО action51;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr6 ТО action53;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr8 ТО action58;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr8 ТО action59;  
ЕСЛИ fact1=fact1attr1 И fact2=fact2attr9 ТО action65;
```













**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr16* **ТО** *action146*;  
**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr16* **ТО** *action147*;  
**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr17* **ТО** *action148*;  
**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr17* **ТО** *action149*;  
**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr17* **ТО** *action150*;  
**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr17* **ТО** *action151*;  
**ЕСЛИ** *fact1=fact1attr3* **И** *fact2=fact2attr17* **ТО** *action152*;

Листинг В.1. Исходный набор продукционных правил

Таблица В.1. Объекты правил (антецедентов)

№ п/п	Объект	Описание	Свойства (атрибуты) объекта
1	2	3	4
1	<b>fact1</b> =Класс защищенности	Класс защищенности АСУ ТП	<b>fact1attr1</b> =3-ий класс защищенности; <b>fact1attr2</b> =2-ой класс защищенности; <b>fact1attr3</b> =1-ый класс защищенности
2	<b>fact2</b> =Группа мер защиты информации в АСУ ТП	Группа мер защиты информации в АСУ ТП	<b>fact2attr1</b> =I. ИАФ; <b>fact2attr2</b> =II. УПД; <b>fact2attr3</b> =III. ОПС; <b>fact2attr4</b> =IV. ЗНИ; <b>fact2attr5</b> =V. АУД; <b>fact2attr6</b> =VI. АВЗ; <b>fact2attr7</b> =VII. СОВ; <b>fact2attr8</b> =VIII. ОЦЛ; <b>fact2attr9</b> =IX. ОДТ; <b>fact2attr10</b> =X. ЗТС; <b>fact2attr11</b> =XI. ЗИС; <b>fact2attr12</b> =XII. ИНЦ; <b>fact2attr13</b> =XIII. УКФ; <b>fact2attr14</b> =XIV. ОПО; <b>fact2attr15</b> =XV. ПЛН; <b>fact2attr16</b> =XVI. ДНС; <b>fact2attr17</b> =XVII. ИПО

В таблице В.1:

- I. ИАФ - Идентификация и аутентификация;
- II. УПД - Управление доступом;
- III. ОПС - Ограничение программной среды;
- IV. ЗНИ - Защита машинных носителей информации;
- V. АУД - Аудит безопасности;
- VI. АВЗ - Антивирусная защита;
- VII. СОВ - Предотвращение вторжений (компьютерных атак);
- VIII. ОЦЛ - Обеспечение целостности;

- IX. ОДТ - Обеспечение доступности;
- X. ЗТС - Защита технических средств и систем;
- XI. ЗИС - Защита информационной (автоматизированной) системы и ее компонентов;
- XII. ИНЦ - Реагирование на компьютерные инциденты;
- XIII. УКФ - Управление конфигурацией;
- XIV. ОПО - Управление обновлениями программного обеспечения;
- XV. ПЛН - Планирование мероприятий по обеспечению безопасности;
- XVI. ДНС - Обеспечение действий в нештатных ситуациях;
- XVII. ИПО - Информирование и обучение персонала.

Таблица В.2. Объекты действий (консеквентов)

№ п/п	Объект	Описание
1	2	3
1	<b>action<sub>1</sub></b> =ИАФ.0	Разработка политики идентификации и аутентификации
2	<b>action<sub>2</sub></b> =ИАФ.1	Идентификация и аутентификация пользователей и инициируемых ими процессов
3	<b>action<sub>3</sub></b> =ИАФ.2	Идентификация и аутентификация устройств
4	<b>action<sub>4</sub></b> =ИАФ.3	Управление идентификаторами
5	<b>action<sub>5</sub></b> =ИАФ.4	Управление средствами аутентификации
6	<b>action<sub>6</sub></b> =ИАФ.5	Идентификация и аутентификация внешних пользователей
7	<b>action<sub>7</sub></b> =ИАФ.6	Двусторонняя аутентификация
8	<b>action<sub>8</sub></b> =ИАФ.7	Защита аутентификационной информации при передаче
9	<b>action<sub>9</sub></b> =УПД.0	Разработка политики управления доступом
10	<b>action<sub>10</sub></b> =УПД.1	Управление учетными записями пользователей
11	<b>action<sub>11</sub></b> =УПД.2	Реализация политик управления доступа
12	<b>action<sub>12</sub></b> =УПД.3	Доверенная загрузка
13	<b>action<sub>13</sub></b> =УПД.4	Разделение полномочий (ролей) пользователей
14	<b>action<sub>14</sub></b> =УПД.5	Назначение минимально необходимых прав и привилегий
15	<b>action<sub>15</sub></b> =УПД.6	Ограничение неуспешных попыток доступа в информационную (автоматизированную) систему
16	<b>action<sub>16</sub></b> =УПД.7	Предупреждение пользователя при его доступе к информационным ресурсам
17	<b>action<sub>17</sub></b> =УПД.8	Оповещение пользователя при успешном входе предыдущем доступе к информационной (автоматизированной) системе
18	<b>action<sub>18</sub></b> =УПД.9	Ограничение числа параллельных сеансов доступа
19	<b>action<sub>19</sub></b> =УПД.10	Блокирование сеанса доступа пользователя при неактивности

<b>№ п/п</b>	<b>Объект</b>	<b>Описание</b>
<b>1</b>	<b>2</b>	<b>3</b>
20	<b>action<sub>20</sub></b> =УПД.11	Управление действиями пользователей до идентификации и аутентификации
21	<b>action<sub>21</sub></b> =УПД.12	Управление атрибутами безопасности
22	<b>action<sub>22</sub></b> =УПД.13	Реализация защищенного удаленного доступа
23	<b>action<sub>23</sub></b> =УПД.14	Контроль доступа из внешних информационных (автоматизированных) систем
24	<b>action<sub>24</sub></b> =ОПС.0	Разработка политики ограничения программной среды
25	<b>action<sub>25</sub></b> =ОПС.1	Управление запуском (обращениями) компонентов программного обеспечения
26	<b>action<sub>26</sub></b> =ОПС.2	Управление установкой (инсталляцией) компонентов программного обеспечения
27	<b>action<sub>27</sub></b> =ОПС.3	Управление временными файлами
28	<b>action<sub>28</sub></b> =ЗНИ.0	Разработка политики защиты машинных носителей информации
29	<b>action<sub>29</sub></b> =ЗНИ.1	Учет машинных носителей информации
30	<b>action<sub>30</sub></b> =ЗНИ.2	Управление физическим доступом к машинным носителям информации
31	<b>action<sub>31</sub></b> =ЗНИ.3	Контроль перемещения машинных носителей информации за пределы контролируемой зоны
32	<b>action<sub>32</sub></b> =ЗНИ.4	Исключение возможности несанкционированного чтения информации на машинных носителях информации
33	<b>action<sub>33</sub></b> =ЗНИ.5	Контроль использования интерфейсов ввода (вывода) информации на машинные носители информации
34	<b>action<sub>34</sub></b> =ЗНИ.6	Контроль ввода (вывода) информации на машинные носители информации
35	<b>action<sub>35</sub></b> =ЗНИ.7	Контроль подключения машинных носителей информации
36	<b>action<sub>36</sub></b> =ЗНИ.8	Уничтожение (стирание) информации на машинных носителях информации
37	<b>action<sub>37</sub></b> =АУД.0	Разработка политики аудита безопасности
38	<b>action<sub>38</sub></b> =АУД.1	Инвентаризация информационных ресурсов
39	<b>action<sub>39</sub></b> =АУД.2	Анализ уязвимостей и их устранение
40	<b>action<sub>40</sub></b> =АУД.3	Генерирование временных меток и (или) синхронизация системного времени
41	<b>action<sub>41</sub></b> =АУД.4	Регистрация событий безопасности
42	<b>action<sub>42</sub></b> =АУД.5	Контроль и анализ сетевого трафика
43	<b>action<sub>43</sub></b> =АУД.6	Защита информации о событиях безопасности
44	<b>action<sub>44</sub></b> =АУД.7	Мониторинг безопасности
45	<b>action<sub>45</sub></b> =АУД.8	Реагирование на сбои при регистрации событий безопасности
46	<b>action<sub>46</sub></b> =АУД.9	Анализ действий пользователей
47	<b>action<sub>47</sub></b> =АУД.10	Проведение внутренних аудитов
48	<b>action<sub>48</sub></b> =АУД.11	Проведение внешних аудитов
49	<b>action<sub>49</sub></b> =АВЗ.0	Разработка политики антивирусной защиты
50	<b>action<sub>50</sub></b> =АВЗ.1	Реализация антивирусной защиты

№ п/п	Объект	Описание
1	2	3
51	action <sub>51</sub> =AB3.2	Антивирусная защита электронной почты и иных сервисов
52	action <sub>52</sub> =AB3.3	Контроль использования архивных, исполняемых и зашифрованных файлов
53	action <sub>53</sub> =AB3.4	Обновление базы данных признаков вредоносных компьютерных программ (вирусов)
54	action <sub>54</sub> =AB3.5	Использование средств антивирусной защиты различных производителей
55	action <sub>55</sub> =COB.0	Разработка политики предотвращения вторжений (компьютерных атак)
56	action <sub>56</sub> =COB.1	Обнаружение и предотвращение компьютерных атак
57	action <sub>57</sub> =COB.2	Обновление базы решающих правил
58	action <sub>58</sub> =ОЦЛ.0	Разработка политики обеспечения целостности
59	action <sub>59</sub> =ОЦЛ.1	Контроль целостности программного обеспечения
60	action <sub>60</sub> =ОЦЛ.2	Контроль целостности информации
61	action <sub>61</sub> =ОЦЛ.3	Ограничения по вводу информации в информационную (автоматизированную) систему
62	action <sub>62</sub> =ОЦЛ.4	Контроль данных, вводимых в информационную (автоматизированную) систему
63	action <sub>63</sub> =ОЦЛ.5	Контроль ошибочных действий пользователей по вводу и (или) передаче информации и предупреждение пользователей об ошибочных действиях
64	action <sub>64</sub> =ОЦЛ.6	Обезличивание и (или) деидентификация информации
65	action <sub>65</sub> =ОДТ.0	Разработка политики обеспечения доступности
66	action <sub>66</sub> =ОДТ.1	Использование отказоустойчивых технических средств
67	action <sub>67</sub> =ОДТ.2	Резервирование средств и систем
68	action <sub>68</sub> =ОДТ.3	Контроль безотказного функционирования средств и систем
69	action <sub>69</sub> =ОДТ.4	Резервное копирование информации
70	action <sub>70</sub> =ОДТ.5	Обеспечение возможности восстановления информации
71	action <sub>71</sub> =ОДТ.6	Обеспечение возможности восстановления программного обеспечения при нештатных ситуациях
72	action <sub>72</sub> =ОДТ.7	Кластеризация информационной (автоматизированной) системы
73	action <sub>73</sub> =ОДТ.8	Контроль предоставляемых вычислительных ресурсов и каналов связи
74	action <sub>74</sub> =ЗТС.0	Разработка политики защиты технических средств и систем
75	action <sub>75</sub> =ЗТС.1	Защита информации от утечки по техническим каналам
76	action <sub>76</sub> =ЗТС.2	Организация контролируемой зоны
77	action <sub>77</sub> =ЗТС.3	Управление физическим доступом
78	action <sub>78</sub> =ЗТС.4	Размещение устройств вывода (отображения) информации, исключающее ее несанкционированный просмотр
79	action <sub>79</sub> =ЗТС.5	Защита от внешних воздействий
80	action <sub>80</sub> =ЗТС.6	Маркирование аппаратных компонентов системы относительно разрешенной к обработке информации

<b>№ п/п</b>	<b>Объект</b>	<b>Описание</b>
<b>1</b>	<b>2</b>	<b>3</b>
81	<b>action<sub>81</sub>=ЗИС.0</b>	Разработка политики защиты информационной (автоматизированной) системы и ее компонентов
82	<b>action<sub>82</sub>=ЗИС.1</b>	Разделение функций по управлению (администрированию) информационной (автоматизированной) системой с иными функциями
83	<b>action<sub>83</sub>=ЗИС.2</b>	Защита периметра информационной (автоматизированной) системы
84	<b>action<sub>84</sub>=ЗИС.3</b>	Эшелонированная защита информационной (автоматизированной) системы
85	<b>action<sub>85</sub>=ЗИС.4</b>	Сегментирование информационной (автоматизированной) системы
86	<b>action<sub>86</sub>=ЗИС.5</b>	Организация демилитаризованной зоны
87	<b>action<sub>87</sub>=ЗИС.6</b>	Управление сетевыми потоками
88	<b>action<sub>88</sub>=ЗИС.7</b>	Использование эмулятора среды функционирования программного обеспечения ("песочница")
89	<b>action<sub>89</sub>=ЗИС.8</b>	Соккрытие архитектуры и конфигурации информационной (автоматизированной) системы
90	<b>action<sub>90</sub>=ЗИС.9</b>	Создание гетерогенной среды
91	<b>action<sub>91</sub>=ЗИС.10</b>	Использование программного обеспечения, функционирующего в средах различных операционных систем
92	<b>action<sub>92</sub>=ЗИС.11</b>	Предотвращение задержки или прерывания выполнения процессов с высоким приоритетом со стороны процессов с низким приоритетом
93	<b>action<sub>93</sub>=ЗИС.12</b>	Изоляция процессов (выполнение программ) в выделенной области памяти
94	<b>action<sub>94</sub>=ЗИС.13</b>	Защита неизменяемых данных
95	<b>action<sub>95</sub>=ЗИС.14</b>	Использование непerezаписываемых машинных носителей информации
96	<b>action<sub>96</sub>=ЗИС.15</b>	Реализация электронного почтового обмена с внешними сетями через ограниченное количество контролируемых точек
97	<b>action<sub>97</sub>=ЗИС.16</b>	Защита от спама
98	<b>action<sub>98</sub>=ЗИС.17</b>	Защита информации от утечек
99	<b>action<sub>99</sub>=ЗИС.18</b>	Блокировка доступа к сайтам или типам сайтов, запрещенных к использованию
100	<b>action<sub>100</sub>=ЗИС.19</b>	Защита информации при ее передаче по каналам связи
101	<b>action<sub>101</sub>=ЗИС.20</b>	Обеспечение доверенных канала, маршрута
102	<b>action<sub>102</sub>=ЗИС.21</b>	Запрет несанкционированной удаленной активации периферийных устройств
103	<b>action<sub>103</sub>=ЗИС.22</b>	Управление атрибутами безопасности при взаимодействии с иными информационными (автоматизированными) системами
104	<b>action<sub>104</sub>=ЗИС.23</b>	Контроль использования мобильного кода
105	<b>action<sub>105</sub>=ЗИС.24</b>	Контроль передачи речевой информации
106	<b>action<sub>106</sub>=ЗИС.25</b>	Контроль передачи видеoinформации
107	<b>action<sub>107</sub>=ЗИС.26</b>	Подтверждение происхождения источника информации
108	<b>action<sub>108</sub>=ЗИС.27</b>	Обеспечение подлинности сетевых соединений

№ п/п	Объект	Описание
1	2	3
109	action <sub>109</sub> =ЗИС.28	Исключение возможности отрицания отправки информации
110	action <sub>110</sub> =ЗИС.29	Исключение возможности отрицания получения информации
111	action <sub>111</sub> =ЗИС.30	Использование устройств терминального доступа
112	action <sub>112</sub> =ЗИС.31	Защита от скрытых каналов передачи информации
113	action <sub>113</sub> =ЗИС.32	Защита беспроводных соединений
114	action <sub>114</sub> =ЗИС.33	Исключение доступа через общие ресурсы
115	action <sub>115</sub> =ЗИС.34	Защита от угроз отказа в обслуживании (DOS, DDOS-атак)
116	action <sub>116</sub> =ЗИС.35	Управление сетевыми соединениями
117	action <sub>117</sub> =ЗИС.36	Создание (эмуляция) ложных компонентов информационных (автоматизированных) систем
118	action <sub>118</sub> =ЗИС.37	Перевод информационной (автоматизированной) системы в безопасное состояние при возникновении отказов (сбоев)
119	action <sub>119</sub> =ЗИС.38	Защита информации при использовании мобильных устройств
120	action <sub>120</sub> =ЗИС.39	Управление перемещением виртуальных машин (контейнеров) и обрабатываемых на них данных
121	action <sub>121</sub> =ИНЦ.0	Разработка политики реагирования на компьютерные инциденты
122	action <sub>122</sub> =ИНЦ.1	Выявление компьютерных инцидентов
123	action <sub>123</sub> =ИНЦ.2	Информирование о компьютерных инцидентах
124	action <sub>124</sub> =ИНЦ.3	Анализ компьютерных инцидентов
125	action <sub>125</sub> =ИНЦ.4	Устранение последствий компьютерных инцидентов
126	action <sub>126</sub> =ИНЦ.5	Принятие мер по предотвращению повторного возникновения компьютерных инцидентов
127	action <sub>127</sub> =ИНЦ.6	Хранение и защита информации о компьютерных инцидентах
128	action <sub>128</sub> =УКФ.0	Разработка политики управления конфигурацией информационной (автоматизированной) системы
129	action <sub>129</sub> =УКФ.1	Идентификация объектов управления конфигурацией
130	action <sub>130</sub> =УКФ.2	Управление изменениями
131	action <sub>131</sub> =УКФ.3	Установка (инсталляция) только разрешенного к использованию программного обеспечения
132	action <sub>132</sub> =УКФ.4	Контроль действий по внесению изменений
133	action <sub>133</sub> =ОПО.0	Разработка политики управления обновлениями программного обеспечения
134	action <sub>134</sub> =ОПО.1	Поиск, получение обновлений программного обеспечения от доверенного источника
135	action <sub>135</sub> =ОПО.2	Контроль целостности обновлений программного обеспечения
136	action <sub>136</sub> =ОПО.3	Тестирование обновлений программного обеспечения
137	action <sub>137</sub> =ОПО.4	Установка обновлений программного обеспечения
138	action <sub>138</sub> =ПЛН.0	Разработка политики планирования мероприятий по обеспечению защиты информации

№ п/п	Объект	Описание
1	2	3
139	action <sub>139</sub> =ПЛН.1	Разработка, утверждение и актуализация плана мероприятий по обеспечению защиты информации
140	action <sub>140</sub> =ПЛН.2	Контроль выполнения мероприятий по обеспечению защиты информации
141	action <sub>141</sub> =ДНС.0	Разработка политики обеспечения действий в нештатных ситуациях
142	action <sub>142</sub> =ДНС.1	Разработка плана действий в нештатных ситуациях
143	action <sub>143</sub> =ДНС.2	Обучение и отработка действий персонала в нештатных ситуациях
144	action <sub>144</sub> =ДНС.3	Создание альтернативных мест хранения и обработки информации на случай возникновения нештатных ситуаций
145	action <sub>145</sub> =ДНС.4	Резервирование программного обеспечения, технических средств, каналов связи на случай возникновения нештатных ситуаций
146	action <sub>146</sub> =ДНС.5	Обеспечение возможности восстановления информационной (автоматизированной) системы в случае возникновения нештатных ситуаций
147	action <sub>147</sub> =ДНС.6	Анализ возникших нештатных ситуаций и принятие мер по недопущению их повторного возникновения
148	action <sub>148</sub> =ИПО.0	Разработка политики информирования и обучения персонала
149	action <sub>149</sub> =ИПО.1	Информирование персонала об угрозах безопасности информации и о правилах безопасной работы
150	action <sub>150</sub> =ИПО.2	Обучение персонала правилам безопасной работы
151	action <sub>151</sub> =ИПО.3	Проведение практических занятий с персоналом по правилам безопасной работы
152	action <sub>152</sub> =ИПО.4	Контроль осведомленности персонала об угрозах безопасности информации и о правилах безопасной работы

В Листинге В.2 представлена БЗ рассматриваемой продукционной экспертной ИС после применения одноитерационного алгоритма извлечения аспектов из последовательности фактов.

```

ЕСЛИ aspect1 ТО action1;
ЕСЛИ aspect1 ТО action2;
ЕСЛИ aspect1 ТО action3;
ЕСЛИ aspect1 ТО action4;
ЕСЛИ aspect1 ТО action5;
ЕСЛИ aspect1 ТО action6;
ЕСЛИ aspect1 ТО action8;
ЕСЛИ aspect2 ТО action9;
ЕСЛИ aspect2 ТО action10;
ЕСЛИ aspect2 ТО action11;
ЕСЛИ aspect2 ТО action13;
ЕСЛИ aspect2 ТО action14;
ЕСЛИ aspect2 ТО action15;

```

ЕСЛИ *aspect2* ТО *action19*;  
ЕСЛИ *aspect2* ТО *action20*;  
ЕСЛИ *aspect2* ТО *action22*;  
ЕСЛИ *aspect2* ТО *action23*;  
ЕСЛИ *aspect3* ТО *action28*;  
ЕСЛИ *aspect3* ТО *action29*;  
ЕСЛИ *aspect3* ТО *action30*;  
ЕСЛИ *aspect3* ТО *action33*;  
ЕСЛИ *aspect3* ТО *action35*;  
ЕСЛИ *aspect3* ТО *action36*;  
ЕСЛИ *aspect4* ТО *action37*;  
ЕСЛИ *aspect4* ТО *action38*;  
ЕСЛИ *aspect4* ТО *action39*;  
ЕСЛИ *aspect4* ТО *action40*;  
ЕСЛИ *aspect4* ТО *action41*;  
ЕСЛИ *aspect4* ТО *action43*;  
ЕСЛИ *aspect4* ТО *action44*;  
ЕСЛИ *aspect4* ТО *action45*;  
ЕСЛИ *aspect4* ТО *action47*;  
ЕСЛИ *aspect5* ТО *action49*;  
ЕСЛИ *aspect5* ТО *action50*;  
ЕСЛИ *aspect5* ТО *action51*;  
ЕСЛИ *aspect5* ТО *action53*;  
ЕСЛИ *aspect6* ТО *action58*;  
ЕСЛИ *aspect6* ТО *action59*;  
ЕСЛИ *aspect7* ТО *action65*;  
ЕСЛИ *aspect7* ТО *action69*;  
ЕСЛИ *aspect7* ТО *action70*;  
ЕСЛИ *aspect7* ТО *action71*;  
ЕСЛИ *aspect7* ТО *action73*;  
ЕСЛИ *aspect8* ТО *action74*;  
ЕСЛИ *aspect8* ТО *action76*;  
ЕСЛИ *aspect8* ТО *action77*;  
ЕСЛИ *aspect8* ТО *action78*;  
ЕСЛИ *aspect8* ТО *action79*;  
ЕСЛИ *aspect9* ТО *action81*;  
ЕСЛИ *aspect9* ТО *action82*;  
ЕСЛИ *aspect9* ТО *action83*;  
ЕСЛИ *aspect9* ТО *action84*;  
ЕСЛИ *aspect9* ТО *action86*;  
ЕСЛИ *aspect9* ТО *action89*;  
ЕСЛИ *aspect9* ТО *action100*;  
ЕСЛИ *aspect9* ТО *action101*;  
ЕСЛИ *aspect9* ТО *action102*;  
ЕСЛИ *aspect9* ТО *action113*;  
ЕСЛИ *aspect9* ТО *action115*;  
ЕСЛИ *aspect9* ТО *action119*;  
ЕСЛИ *aspect9* ТО *action120*;  
ЕСЛИ *aspect10* ТО *action121*;  
ЕСЛИ *aspect10* ТО *action122*;  
ЕСЛИ *aspect10* ТО *action123*;  
ЕСЛИ *aspect10* ТО *action124*;

ЕСЛИ *aspect10* **TO** *action125*;  
ЕСЛИ *aspect10* **TO** *action126*;  
ЕСЛИ *aspect11* **TO** *action128*;  
ЕСЛИ *aspect11* **TO** *action130*;  
ЕСЛИ *aspect11* **TO** *action131*;  
ЕСЛИ *aspect12* **TO** *action133*;  
ЕСЛИ *aspect12* **TO** *action134*;  
ЕСЛИ *aspect12* **TO** *action135*;  
ЕСЛИ *aspect12* **TO** *action136*;  
ЕСЛИ *aspect12* **TO** *action137*;  
ЕСЛИ *aspect13* **TO** *action138*;  
ЕСЛИ *aspect13* **TO** *action139*;  
ЕСЛИ *aspect13* **TO** *action140*;  
ЕСЛИ *aspect14* **TO** *action141*;  
ЕСЛИ *aspect14* **TO** *action142*;  
ЕСЛИ *aspect14* **TO** *action143*;  
ЕСЛИ *aspect14* **TO** *action146*;  
ЕСЛИ *aspect14* **TO** *action147*;  
ЕСЛИ *aspect15* **TO** *action148*;  
ЕСЛИ *aspect15* **TO** *action149*;  
ЕСЛИ *aspect15* **TO** *action150*;  
ЕСЛИ *aspect15* **TO** *action152*;  
ЕСЛИ *aspect16* **TO** *action1*;  
ЕСЛИ *aspect16* **TO** *action2*;  
ЕСЛИ *aspect16* **TO** *action3*;  
ЕСЛИ *aspect16* **TO** *action4*;  
ЕСЛИ *aspect16* **TO** *action5*;  
ЕСЛИ *aspect16* **TO** *action6*;  
ЕСЛИ *aspect16* **TO** *action8*;  
ЕСЛИ *aspect17* **TO** *action9*;  
ЕСЛИ *aspect17* **TO** *action10*;  
ЕСЛИ *aspect17* **TO** *action11*;  
ЕСЛИ *aspect17* **TO** *action12*;  
ЕСЛИ *aspect17* **TO** *action13*;  
ЕСЛИ *aspect17* **TO** *action14*;  
ЕСЛИ *aspect17* **TO** *action15*;  
ЕСЛИ *aspect17* **TO** *action19*;  
ЕСЛИ *aspect17* **TO** *action20*;  
ЕСЛИ *aspect17* **TO** *action22*;  
ЕСЛИ *aspect17* **TO** *action23*;  
ЕСЛИ *aspect18* **TO** *action24*;  
ЕСЛИ *aspect18* **TO** *action26*;  
ЕСЛИ *aspect19* **TO** *action28*;  
ЕСЛИ *aspect19* **TO** *action29*;  
ЕСЛИ *aspect19* **TO** *action30*;  
ЕСЛИ *aspect19* **TO** *action33*;  
ЕСЛИ *aspect19* **TO** *action35*;  
ЕСЛИ *aspect19* **TO** *action36*;  
ЕСЛИ *aspect20* **TO** *action37*;  
ЕСЛИ *aspect20* **TO** *action38*;  
ЕСЛИ *aspect20* **TO** *action39*;  
ЕСЛИ *aspect20* **TO** *action40*;

ЕСЛИ *aspect20* **TO** *action41*;  
ЕСЛИ *aspect20* **TO** *action43*;  
ЕСЛИ *aspect20* **TO** *action44*;  
ЕСЛИ *aspect20* **TO** *action45*;  
ЕСЛИ *aspect20* **TO** *action47*;  
ЕСЛИ *aspect21* **TO** *action49*;  
ЕСЛИ *aspect21* **TO** *action50*;  
ЕСЛИ *aspect21* **TO** *action51*;  
ЕСЛИ *aspect21* **TO** *action53*;  
ЕСЛИ *aspect22* **TO** *action55*;  
ЕСЛИ *aspect22* **TO** *action56*;  
ЕСЛИ *aspect22* **TO** *action57*;  
ЕСЛИ *aspect23* **TO** *action58*;  
ЕСЛИ *aspect23* **TO** *action59*;  
ЕСЛИ *aspect23* **TO** *action62*;  
ЕСЛИ *aspect23* **TO** *action63*;  
ЕСЛИ *aspect24* **TO** *action65*;  
ЕСЛИ *aspect24* **TO** *action66*;  
ЕСЛИ *aspect24* **TO** *action67*;  
ЕСЛИ *aspect24* **TO** *action68*;  
ЕСЛИ *aspect24* **TO** *action69*;  
ЕСЛИ *aspect24* **TO** *action70*;  
ЕСЛИ *aspect24* **TO** *action71*;  
ЕСЛИ *aspect24* **TO** *action73*;  
ЕСЛИ *aspect25* **TO** *action74*;  
ЕСЛИ *aspect25* **TO** *action76*;  
ЕСЛИ *aspect25* **TO** *action77*;  
ЕСЛИ *aspect25* **TO** *action78*;  
ЕСЛИ *aspect25* **TO** *action79*;  
ЕСЛИ *aspect26* **TO** *action81*;  
ЕСЛИ *aspect26* **TO** *action82*;  
ЕСЛИ *aspect26* **TO** *action83*;  
ЕСЛИ *aspect26* **TO** *action84*;  
ЕСЛИ *aspect26* **TO** *action85*;  
ЕСЛИ *aspect26* **TO** *action86*;  
ЕСЛИ *aspect26* **TO** *action89*;  
ЕСЛИ *aspect26* **TO** *action94*;  
ЕСЛИ *aspect26* **TO** *action97*;  
ЕСЛИ *aspect26* **TO** *action100*;  
ЕСЛИ *aspect26* **TO** *action101*;  
ЕСЛИ *aspect26* **TO** *action102*;  
ЕСЛИ *aspect26* **TO** *action104*;  
ЕСЛИ *aspect26* **TO** *action105*;  
ЕСЛИ *aspect26* **TO** *action106*;  
ЕСЛИ *aspect26* **TO** *action108*;  
ЕСЛИ *aspect26* **TO** *action109*;  
ЕСЛИ *aspect26* **TO** *action110*;  
ЕСЛИ *aspect26* **TO** *action113*;  
ЕСЛИ *aspect26* **TO** *action115*;  
ЕСЛИ *aspect26* **TO** *action116*;  
ЕСЛИ *aspect26* **TO** *action119*;  
ЕСЛИ *aspect26* **TO** *action120*;

ЕСЛИ *aspect27* **TO** *action121*;  
ЕСЛИ *aspect27* **TO** *action122*;  
ЕСЛИ *aspect27* **TO** *action123*;  
ЕСЛИ *aspect27* **TO** *action124*;  
ЕСЛИ *aspect27* **TO** *action125*;  
ЕСЛИ *aspect27* **TO** *action126*;  
ЕСЛИ *aspect28* **TO** *action128*;  
ЕСЛИ *aspect28* **TO** *action130*;  
ЕСЛИ *aspect28* **TO** *action131*;  
ЕСЛИ *aspect29* **TO** *action133*;  
ЕСЛИ *aspect29* **TO** *action134*;  
ЕСЛИ *aspect29* **TO** *action135*;  
ЕСЛИ *aspect29* **TO** *action136*;  
ЕСЛИ *aspect29* **TO** *action137*;  
ЕСЛИ *aspect30* **TO** *action138*;  
ЕСЛИ *aspect30* **TO** *action139*;  
ЕСЛИ *aspect30* **TO** *action140*;  
ЕСЛИ *aspect31* **TO** *action141*;  
ЕСЛИ *aspect31* **TO** *action142*;  
ЕСЛИ *aspect31* **TO** *action143*;  
ЕСЛИ *aspect31* **TO** *action144*;  
ЕСЛИ *aspect31* **TO** *action145*;  
ЕСЛИ *aspect31* **TO** *action146*;  
ЕСЛИ *aspect31* **TO** *action147*;  
ЕСЛИ *aspect32* **TO** *action148*;  
ЕСЛИ *aspect32* **TO** *action149*;  
ЕСЛИ *aspect32* **TO** *action150*;  
ЕСЛИ *aspect32* **TO** *action151*;  
ЕСЛИ *aspect32* **TO** *action152*;  
ЕСЛИ *aspect33* **TO** *action1*;  
ЕСЛИ *aspect33* **TO** *action2*;  
ЕСЛИ *aspect33* **TO** *action3*;  
ЕСЛИ *aspect33* **TO** *action4*;  
ЕСЛИ *aspect33* **TO** *action5*;  
ЕСЛИ *aspect33* **TO** *action6*;  
ЕСЛИ *aspect33* **TO** *action8*;  
ЕСЛИ *aspect34* **TO** *action9*;  
ЕСЛИ *aspect34* **TO** *action10*;  
ЕСЛИ *aspect34* **TO** *action11*;  
ЕСЛИ *aspect34* **TO** *action12*;  
ЕСЛИ *aspect34* **TO** *action13*;  
ЕСЛИ *aspect34* **TO** *action14*;  
ЕСЛИ *aspect34* **TO** *action15*;  
ЕСЛИ *aspect34* **TO** *action17*;  
ЕСЛИ *aspect34* **TO** *action18*;  
ЕСЛИ *aspect34* **TO** *action19*;  
ЕСЛИ *aspect34* **TO** *action20*;  
ЕСЛИ *aspect34* **TO** *action22*;  
ЕСЛИ *aspect34* **TO** *action23*;  
ЕСЛИ *aspect35* **TO** *action24*;  
ЕСЛИ *aspect35* **TO** *action25*;  
ЕСЛИ *aspect35* **TO** *action26*;

ЕСЛИ *aspect36* **TO** *action28*;  
ЕСЛИ *aspect36* **TO** *action29*;  
ЕСЛИ *aspect36* **TO** *action30*;  
ЕСЛИ *aspect36* **TO** *action33*;  
ЕСЛИ *aspect36* **TO** *action34*;  
ЕСЛИ *aspect36* **TO** *action35*;  
ЕСЛИ *aspect36* **TO** *action36*;  
ЕСЛИ *aspect37* **TO** *action37*;  
ЕСЛИ *aspect37* **TO** *action38*;  
ЕСЛИ *aspect37* **TO** *action39*;  
ЕСЛИ *aspect37* **TO** *action40*;  
ЕСЛИ *aspect37* **TO** *action41*;  
ЕСЛИ *aspect37* **TO** *action42*;  
ЕСЛИ *aspect37* **TO** *action43*;  
ЕСЛИ *aspect37* **TO** *action44*;  
ЕСЛИ *aspect37* **TO** *action45*;  
ЕСЛИ *aspect37* **TO** *action46*;  
ЕСЛИ *aspect37* **TO** *action47*;  
ЕСЛИ *aspect37* **TO** *action48*;  
ЕСЛИ *aspect38* **TO** *action49*;  
ЕСЛИ *aspect38* **TO** *action50*;  
ЕСЛИ *aspect38* **TO** *action51*;  
ЕСЛИ *aspect38* **TO** *action52*;  
ЕСЛИ *aspect38* **TO** *action53*;  
ЕСЛИ *aspect38* **TO** *action54*;  
ЕСЛИ *aspect39* **TO** *action55*;  
ЕСЛИ *aspect39* **TO** *action56*;  
ЕСЛИ *aspect39* **TO** *action57*;  
ЕСЛИ *aspect40* **TO** *action58*;  
ЕСЛИ *aspect40* **TO** *action59*;  
ЕСЛИ *aspect40* **TO** *action61*;  
ЕСЛИ *aspect40* **TO** *action62*;  
ЕСЛИ *aspect40* **TO** *action63*;  
ЕСЛИ *aspect41* **TO** *action65*;  
ЕСЛИ *aspect41* **TO** *action66*;  
ЕСЛИ *aspect41* **TO** *action67*;  
ЕСЛИ *aspect41* **TO** *action68*;  
ЕСЛИ *aspect41* **TO** *action69*;  
ЕСЛИ *aspect41* **TO** *action70*;  
ЕСЛИ *aspect41* **TO** *action71*;  
ЕСЛИ *aspect41* **TO** *action73*;  
ЕСЛИ *aspect42* **TO** *action74*;  
ЕСЛИ *aspect42* **TO** *action76*;  
ЕСЛИ *aspect42* **TO** *action77*;  
ЕСЛИ *aspect42* **TO** *action78*;  
ЕСЛИ *aspect42* **TO** *action79*;  
ЕСЛИ *aspect43* **TO** *action81*;  
ЕСЛИ *aspect43* **TO** *action82*;  
ЕСЛИ *aspect43* **TO** *action83*;  
ЕСЛИ *aspect43* **TO** *action84*;  
ЕСЛИ *aspect43* **TO** *action85*;  
ЕСЛИ *aspect43* **TO** *action86*;

```
ЕСЛИ aspect43 TO action89;
ЕСЛИ aspect43 TO action94;
ЕСЛИ aspect43 TO action97;
ЕСЛИ aspect43 TO action100;
ЕСЛИ aspect43 TO action101;
ЕСЛИ aspect43 TO action102;
ЕСЛИ aspect43 TO action104;
ЕСЛИ aspect43 TO action105;
ЕСЛИ aspect43 TO action106;
ЕСЛИ aspect43 TO action108;
ЕСЛИ aspect43 TO action109;
ЕСЛИ aspect43 TO action110;
ЕСЛИ aspect43 TO action112;
ЕСЛИ aspect43 TO action113;
ЕСЛИ aspect43 TO action114;
ЕСЛИ aspect43 TO action115;
ЕСЛИ aspect43 TO action116;
ЕСЛИ aspect43 TO action119;
ЕСЛИ aspect43 TO action120;
ЕСЛИ aspect44 TO action121;
ЕСЛИ aspect44 TO action122;
ЕСЛИ aspect44 TO action123;
ЕСЛИ aspect44 TO action124;
ЕСЛИ aspect44 TO action125;
ЕСЛИ aspect44 TO action126;
ЕСЛИ aspect44 TO action127;
ЕСЛИ aspect45 TO action128;
ЕСЛИ aspect45 TO action130;
ЕСЛИ aspect45 TO action131;
ЕСЛИ aspect46 TO action133;
ЕСЛИ aspect46 TO action134;
ЕСЛИ aspect46 TO action135;
ЕСЛИ aspect46 TO action136;
ЕСЛИ aspect46 TO action137;
ЕСЛИ aspect47 TO action138;
ЕСЛИ aspect47 TO action139;
ЕСЛИ aspect47 TO action140;
ЕСЛИ aspect48 TO action141;
ЕСЛИ aspect48 TO action142;
ЕСЛИ aspect48 TO action143;
ЕСЛИ aspect48 TO action144;
ЕСЛИ aspect48 TO action145;
ЕСЛИ aspect48 TO action146;
ЕСЛИ aspect48 TO action147;
ЕСЛИ aspect49 TO action148;
ЕСЛИ aspect49 TO action149;
ЕСЛИ aspect49 TO action150;
ЕСЛИ aspect49 TO action151;
ЕСЛИ aspect49 TO action152;
aspect1 = [fact1=fact1attr1 И fact2=fact2attr1]
aspect2 = [fact1=fact1attr1 И fact2=fact2attr2];
aspect3 = [fact1=fact1attr1 И fact2=fact2attr4];
```

```
aspect4 = [fact1=fact1attr1 И fact2=fact2attr5];
aspect5 = [fact1=fact1attr1 И fact2=fact2attr6];
aspect6 = [fact1=fact1attr1 И fact2=fact2attr8];
aspect7 = [fact1=fact1attr1 И fact2=fact2attr9];
aspect8 = [fact1=fact1attr1 И fact2=fact2attr10];
aspect9 = [fact1=fact1attr1 И fact2=fact2attr11];
aspect10 = [fact1=fact1attr1 И fact2=fact2attr12];
aspect11 = [fact1=fact1attr1 И fact2=fact2attr13];
aspect12 = [fact1=fact1attr1 И fact2=fact2attr14];
aspect13 = [fact1=fact1attr1 И fact2=fact2attr15];
aspect14 = [fact1=fact1attr1 И fact2=fact2attr16];
aspect15 = [fact1=fact1attr1 И fact2=fact2attr17];
aspect16 = [fact1=fact1attr2 И fact2=fact2attr1];
aspect17 = [fact1=fact1attr2 И fact2=fact2attr2];
aspect18 = [fact1=fact1attr2 И fact2=fact2attr3];
aspect19 = [fact1=fact1attr2 И fact2=fact2attr4];
aspect20 = [fact1=fact1attr2 И fact2=fact2attr5];
aspect21 = [fact1=fact1attr2 И fact2=fact2attr6];
aspect22 = [fact1=fact1attr2 И fact2=fact2attr7];
aspect23 = [fact1=fact1attr2 И fact2=fact2attr8];
aspect24 = [fact1=fact1attr2 И fact2=fact2attr9];
aspect25 = [fact1=fact1attr2 И fact2=fact2attr10];
aspect26 = [fact1=fact1attr2 И fact2=fact2attr11];
aspect27 = [fact1=fact1attr2 И fact2=fact2attr12];
aspect28 = [fact1=fact1attr2 И fact2=fact2attr13];
aspect29 = [fact1=fact1attr2 И fact2=fact2attr14];
aspect30 = [fact1=fact1attr2 И fact2=fact2attr15];
aspect31 = [fact1=fact1attr2 И fact2=fact2attr16];
aspect32 = [fact1=fact1attr2 И fact2=fact2attr17];
aspect33 = [fact1=fact1attr3 И fact2=fact2attr1];
aspect34 = [fact1=fact1attr3 И fact2=fact2attr2];
aspect35 = [fact1=fact1attr3 И fact2=fact2attr3];
aspect36 = [fact1=fact1attr3 И fact2=fact2attr4];
aspect37 = [fact1=fact1attr3 И fact2=fact2attr5];
aspect38 = [fact1=fact1attr3 И fact2=fact2attr6];
aspect39 = [fact1=fact1attr3 И fact2=fact2attr7];
aspect40 = [fact1=fact1attr3 И fact2=fact2attr8];
aspect41 = [fact1=fact1attr3 И fact2=fact2attr9];
aspect42 = [fact1=fact1attr3 И fact2=fact2attr10];
aspect43 = [fact1=fact1attr3 И fact2=fact2attr11];
aspect44 = [fact1=fact1attr3 И fact2=fact2attr12];
aspect45 = [fact1=fact1attr3 И fact2=fact2attr13];
aspect46 = [fact1=fact1attr3 И fact2=fact2attr14];
aspect47 = [fact1=fact1attr3 И fact2=fact2attr15];
aspect48 = [fact1=fact1attr3 И fact2=fact2attr16];
aspect49 = [fact1=fact1attr3 И fact2=fact2attr17];
```

Листинг В.2. БЗ рассматриваемой продукционной экспертной ИС после применения одноитерационного алгоритма извлечения аспектов из последовательности фактов

В Листинге В.3 представлена БЗ рассматриваемой продукционной экспертной ИС после применения рекурсивного алгоритма извлечения аспектов из последовательности фактов.

```
ЕСЛИ aspect50 ТО action1 И action2 И action3 И action4 И action5
И action6 И action8;
ЕСЛИ aspect51 ТО action9 И action10 И action11 И action13 И
action14 И action15 И action19 И action20 И action22 И action23;
ЕСЛИ aspect52 ТО action28 И action29 И action30 И action33 И
action35 И action36;
ЕСЛИ aspect53 ТО action37 И action38 И action39 И action40 И
action41 И action43 И action44 И action45 И action47;
ЕСЛИ aspect54 ТО action49 И action50 И action51 И action53;
ЕСЛИ aspect55 ТО action58 И action59;
ЕСЛИ aspect56 ТО action65 И action69 И action70 И action71 И
action73;
ЕСЛИ aspect57 ТО action74 И action76 И action77 И action78 И
action79;
ЕСЛИ aspect58 ТО action81 И action82 И action83 И action84 И
action86 И action89 И action100 И action101 И action102 И action113 И
action115 И action119 И action120;
ЕСЛИ aspect59 ТО action121 И action122 И action123 И action124 И
action125 И action126;
ЕСЛИ aspect60 ТО action128 И action130 И action131;
ЕСЛИ aspect61 ТО action133 И action134 И action135 И action136 И
action137;
ЕСЛИ aspect62 ТО action138 И action139 И action140;
ЕСЛИ aspect63 ТО action141 И action142 И action143 И action146 И
action147;
ЕСЛИ aspect64 ТО action148 И action149 И action150 И action152;
ЕСЛИ aspect65 ТО action1 И action2 И action3 И action4 И action5
И action6 И action8;
ЕСЛИ aspect66 ТО action9 И action10 И action11 И action12 И
action13 И action14 И action15 И action19 И action20 И action22 И
action23;
ЕСЛИ aspect67 ТО action24 И action26;
ЕСЛИ aspect68 ТО action28 И action29 И action30 И action33 И
action35 И action36;
ЕСЛИ aspect69 ТО action37 И action38 И action39 И action40 И
action41 И action43 И action44 И action45 И action47;
ЕСЛИ aspect70 ТО action49 И action50 И action51 И action53;
ЕСЛИ aspect71 ТО action55 И action56 И action57;
ЕСЛИ aspect72 ТО action58 И action59 И action62 И action63;
ЕСЛИ aspect73 ТО action65 И action66 И action67 И action68 И
action69 И action70 И action71 И action73;
ЕСЛИ aspect74 ТО action74 И action76 И action77 И action78 И
action79;
ЕСЛИ aspect75 ТО action81 И action82 И action83 И action84 И
action85 И action86 И action89 И action94 И action97 И action100 И
action101 И action102 И action104 И action105 И action106 И action108
```

И action109 И action110 И action113 И action115 И action116 И action119  
И action120;

ЕСЛИ aspect76 ТО action121 И action122 И action123 И action124 И  
action125 И action126;

ЕСЛИ aspect77 ТО action128 И action130 И action131;

ЕСЛИ aspect78 ТО action133 И action134 И action135 И action136 И  
action137;

ЕСЛИ aspect79 ТО action138 И action139 И action140;

ЕСЛИ aspect80 ТО action141 И action142 И action143 И action144 И  
action145 И action146 И action147;

ЕСЛИ aspect81 ТО action148 И action149 И action150 И action151 И  
action152;

ЕСЛИ aspect82 ТО action1 И action2 И action3 И action4 И action5  
И action6 И action8;

ЕСЛИ aspect83 ТО action9 И action10 И action11 И action12 И  
action13 И action14 И action15 И action17 И action18 И action19 И  
action20 И action22 И action23;

ЕСЛИ aspect84 ТО action24 И action25 И action26;

ЕСЛИ aspect85 ТО action28 И action29 И action30 И action33 И  
action34 И action35 И action36;

ЕСЛИ aspect86 ТО action37 И action38 И action39 И action40 И  
action41 И action42 И action43 И action44 И action45 И action46 И  
action47 И action48;

ЕСЛИ aspect87 ТО action49 И action50 И action51 И action52 И  
action53 И action54;

ЕСЛИ aspect88 ТО action55 И action56 И action57;

ЕСЛИ aspect89 ТО action58 И action59 И action61 И action62 И  
action63;

ЕСЛИ aspect90 ТО action65 И action66 И action67 И action68 И  
action69 И action70 И action71 И action73;

ЕСЛИ aspect91 ТО action74 И action76 И action77 И action78 И  
action79;

ЕСЛИ aspect92 ТО action81 И action82 И action83 И action84 И  
action85 И action86 И action89 И action94 И action97 И action100 И  
action101 И action102 И action104 И action105 И action106 И action108  
И action109 И action110 И action112 И action113 И action114 И action115  
И action116 И action119 И action120;

ЕСЛИ aspect93 ТО action121 И action122 И action123 И action124 И  
action125 И action126 И action127;

ЕСЛИ aspect94 ТО action128 И action130 И action131;

ЕСЛИ aspect95 ТО action133 И action134 И action135 И action136 И  
action137;

ЕСЛИ aspect96 ТО action138 И action139 И action140;

ЕСЛИ aspect97 ТО action141 И action142 И action143 И action144 И  
action145 И action146 И action147;

ЕСЛИ aspect98 ТО action148 И action149 И action150 И action151 И  
action152;

aspect1 = [fact1=fact1attr1 И fact2=fact2attr1]  
aspect2 = [fact1=fact1attr1 И fact2=fact2attr2];  
aspect3 = [fact1=fact1attr1 И fact2=fact2attr4];  
aspect4 = [fact1=fact1attr1 И fact2=fact2attr5];  
aspect5 = [fact1=fact1attr1 И fact2=fact2attr6];

```

aspect6 = [fact1=fact1attr1 И fact2=fact2attr8];
aspect7 = [fact1=fact1attr1 И fact2=fact2attr9];
aspect8 = [fact1=fact1attr1 И fact2=fact2attr10];
aspect9 = [fact1=fact1attr1 И fact2=fact2attr11];
aspect10 = [fact1=fact1attr1 И fact2=fact2attr12];
aspect11 = [fact1=fact1attr1 И fact2=fact2attr13];
aspect12 = [fact1=fact1attr1 И fact2=fact2attr14];
aspect13 = [fact1=fact1attr1 И fact2=fact2attr15];
aspect14 = [fact1=fact1attr1 И fact2=fact2attr16];
aspect15 = [fact1=fact1attr1 И fact2=fact2attr17];
aspect16 = [fact1=fact1attr2 И fact2=fact2attr1];
aspect17 = [fact1=fact1attr2 И fact2=fact2attr2];
aspect18 = [fact1=fact1attr2 И fact2=fact2attr3];
aspect19 = [fact1=fact1attr2 И fact2=fact2attr4];
aspect20 = [fact1=fact1attr2 И fact2=fact2attr5];
aspect21 = [fact1=fact1attr2 И fact2=fact2attr6];
aspect22 = [fact1=fact1attr2 И fact2=fact2attr7];
aspect23 = [fact1=fact1attr2 И fact2=fact2attr8];
aspect24 = [fact1=fact1attr2 И fact2=fact2attr9];
aspect25 = [fact1=fact1attr2 И fact2=fact2attr10];
aspect26 = [fact1=fact1attr2 И fact2=fact2attr11];
aspect27 = [fact1=fact1attr2 И fact2=fact2attr12];
aspect28 = [fact1=fact1attr2 И fact2=fact2attr13];
aspect29 = [fact1=fact1attr2 И fact2=fact2attr14];
aspect30 = [fact1=fact1attr2 И fact2=fact2attr15];
aspect31 = [fact1=fact1attr2 И fact2=fact2attr16];
aspect32 = [fact1=fact1attr2 И fact2=fact2attr17];
aspect33 = [fact1=fact1attr3 И fact2=fact2attr1];
aspect34 = [fact1=fact1attr3 И fact2=fact2attr2];
aspect35 = [fact1=fact1attr3 И fact2=fact2attr3];
aspect36 = [fact1=fact1attr3 И fact2=fact2attr4];
aspect37 = [fact1=fact1attr3 И fact2=fact2attr5];
aspect38 = [fact1=fact1attr3 И fact2=fact2attr6];
aspect39 = [fact1=fact1attr3 И fact2=fact2attr7];
aspect40 = [fact1=fact1attr3 И fact2=fact2attr8];
aspect41 = [fact1=fact1attr3 И fact2=fact2attr9];
aspect42 = [fact1=fact1attr3 И fact2=fact2attr10];
aspect43 = [fact1=fact1attr3 И fact2=fact2attr11];
aspect44 = [fact1=fact1attr3 И fact2=fact2attr12];
aspect45 = [fact1=fact1attr3 И fact2=fact2attr13];
aspect46 = [fact1=fact1attr3 И fact2=fact2attr14];
aspect47 = [fact1=fact1attr3 И fact2=fact2attr15];
aspect48 = [fact1=fact1attr3 И fact2=fact2attr16];
aspect49 = [fact1=fact1attr3 И fact2=fact2attr17];
aspect50 = _[aspect1];
aspect51 = _[aspect2];
aspect52 = _[aspect3];
aspect53 = _[aspect4];
aspect54 = _[aspect5];
aspect55 = _[aspect6];
aspect56 = _[aspect7];
aspect57 = _[aspect8];

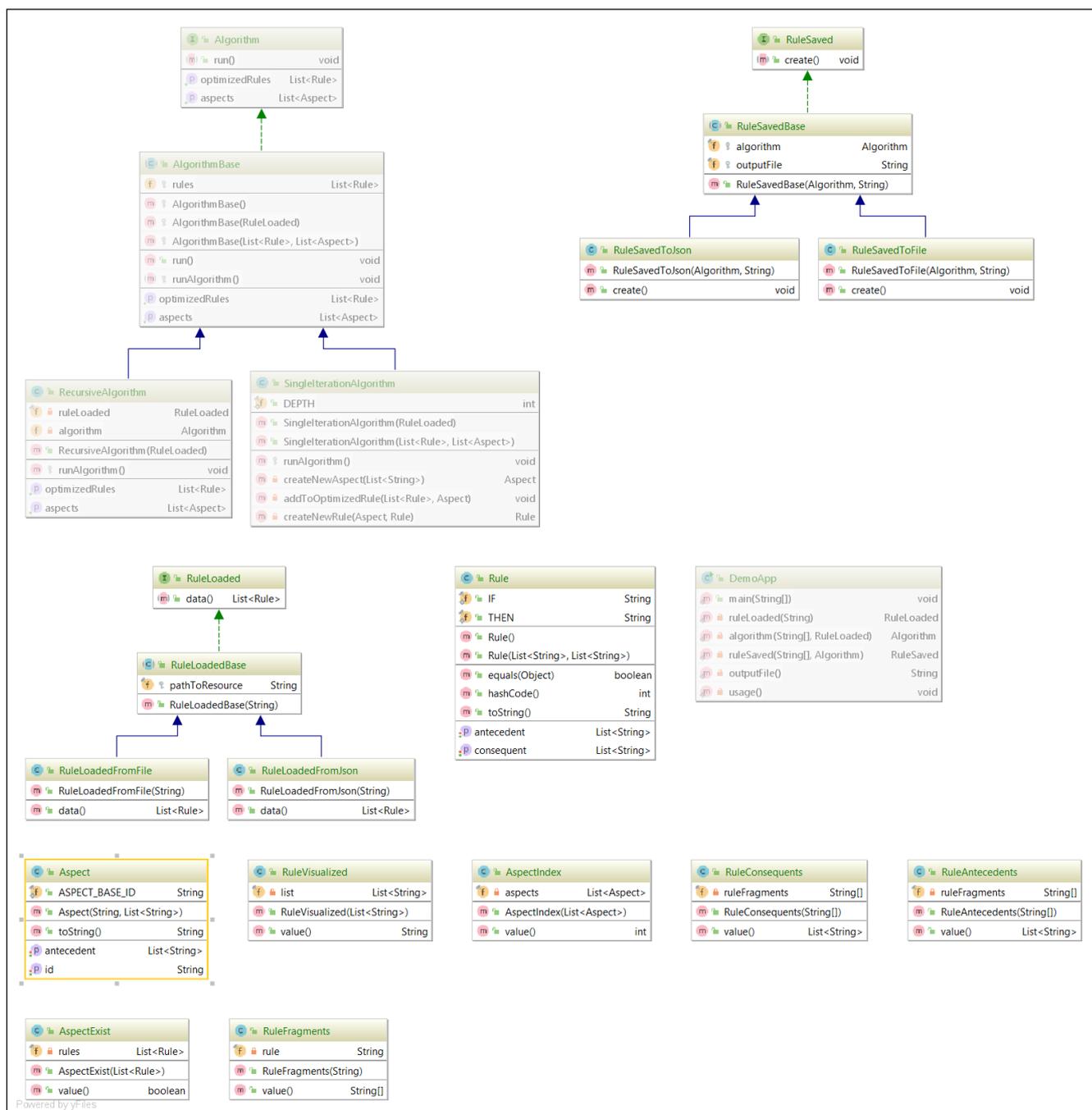
```

```
aspect58 = _[aspect9];
aspect59 = _[aspect10];
aspect60 = _[aspect11];
aspect61 = _[aspect12];
aspect62 = _[aspect13];
aspect63 = _[aspect14];
aspect64 = _[aspect15];
aspect65 = _[aspect16];
aspect66 = _[aspect17];
aspect67 = _[aspect18];
aspect68 = _[aspect19];
aspect69 = _[aspect20];
aspect70 = _[aspect21];
aspect71 = _[aspect22];
aspect72 = _[aspect23];
aspect73 = _[aspect24];
aspect74 = _[aspect25];
aspect75 = _[aspect26];
aspect76 = _[aspect27];
aspect77 = _[aspect28];
aspect78 = _[aspect29];
aspect79 = _[aspect30];
aspect80 = _[aspect31];
aspect81 = _[aspect32];
aspect82 = _[aspect33];
aspect83 = _[aspect34];
aspect84 = _[aspect35];
aspect85 = _[aspect36];
aspect86 = _[aspect37];
aspect87 = _[aspect38];
aspect88 = _[aspect39];
aspect89 = _[aspect40];
aspect90 = _[aspect41];
aspect91 = _[aspect42];
aspect92 = _[aspect43];
aspect93 = _[aspect44];
aspect94 = _[aspect45];
aspect95 = _[aspect46];
aspect96 = _[aspect47];
aspect97 = _[aspect48];
aspect98 = _[aspect49];
```

Листинг В.3. БЗ рассматриваемой продукционной экспертной ИС после применения рекурсивного алгоритма извлечения аспектов из последовательности фактов

## ПРИЛОЖЕНИЕ Б

На рисунке представлена блок-схема взаимодействия классов реализации подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов, построенная с помощью встроенных средств среды разработки IntelliJ IDEA.



## ПРИЛОЖЕНИЕ В

В Приложении В представлен исходный код подсистемы интерпретации продукционных правил экспертной ИС с применением АО-подхода и разработанных алгоритмов извлечения аспектов из последовательности фактов на языке Java.

\src\main\java\ru\goncharov\dissertation\DemoApp.java

```
package ru.goncharov.dissertation;

import ru.goncharov.dissertation.algorithms.Algorithm;
import ru.goncharov.dissertation.algorithms.RecursiveAlgorithm;
import ru.goncharov.dissertation.algorithms.SingleIterationAlgorithm;
import ru.goncharov.dissertation.algorithms.common.*;

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class DemoApp {

    public static void main(String[] args) {
        if (args.length > 0) {
            final RuleLoaded ruleLoaded = ruleLoaded(args[0]);
            final Algorithm algorithm = algorithm(args, ruleLoaded);
            algorithm.run();
            final RuleSaved ruleSaved = ruleSaved(args, algorithm);
            ruleSaved.create();
        } else {
            usage();
        }
    }

    private static RuleLoaded ruleLoaded(final String pathToResource) {
        return !pathToResource.endsWith(".json") ?
            new RuleLoadedFromFile(pathToResource) :
            new RuleLoadedFromJson(pathToResource);
    }

    private static Algorithm algorithm(final String[] args, final RuleLoaded
ruleLoaded) {
        final Algorithm algorithm;
        if (args.length == 2 && "r".equals(args[1]) || args.length == 3 &&
"r".equals(args[1])) {
            System.err.println("> RecursiveAlgorithm");
            algorithm = new RecursiveAlgorithm(ruleLoaded);
        } else {
            System.err.println("> SingleIterationAlgorithm");
            algorithm = new SingleIterationAlgorithm(ruleLoaded);
        }
        return algorithm;
    }

    private static RuleSaved ruleSaved(final String[] args, final Algorithm
algorithm) {
        final String outputFile = outputFile();
        final RuleSaved ruleSaved;
        if (args.length == 2 && "json".equals(args[1]) || args.length == 3 &&
"json".equals(args[2])) {
```

```

        System.err.println("> Json");
        ruleSaved = new RuleSavedToJson(algorithm, outputFile);
    } else {
        System.err.println("> File");
        ruleSaved = new RuleSavedToFile(algorithm, outputFile);
    }
    return ruleSaved;
}

private static String outputFile() {
    return String.format(
        "output_%s",
        DateTimeFormatter.ofPattern("MM_dd_yyyy_HH_mm_ss")
            .format(
                LocalDateTime.now()
            )
    );
}

private static void usage() {
    System.err.println(
        "Usage: java -jar DissertationAlgorithmsDemoApp-1.0-jar-with-
dependencies.jar \"/path/to/resource\" [r] [json]"
    );
}
}

```

`\src\main\java\ru\goncharov\dissertation\algorithms\Algorithm.java`

```

package ru.goncharov.dissertation.algorithms;

import ru.goncharov.dissertation.algorithms.common.Aspect;
import ru.goncharov.dissertation.algorithms.common.Rule;

import java.util.List;

public interface Algorithm {
    void run();
    List<Rule> getOptimizedRules();
    List<Aspect> getAspects();
}

```

`\src\main\java\ru\goncharov\dissertation\algorithms\AlgorithmBase.java`

```

package ru.goncharov.dissertation.algorithms;

import ru.goncharov.dissertation.algorithms.common.Aspect;
import ru.goncharov.dissertation.algorithms.common.Rule;
import ru.goncharov.dissertation.algorithms.common.RuleLoaded;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.CopyOnWriteArrayList;

public abstract class AlgorithmBase implements Algorithm {

    protected List<Rule> rules;
    protected final List<Rule> optimizedRules = new CopyOnWriteArrayList<>();
    protected List<Aspect> aspects;

    protected AlgorithmBase() {
    }

    protected AlgorithmBase(final RuleLoaded ruleLoaded) {
        this.rules = ruleLoaded.data();
    }
}

```

```

        aspects = new ArrayList<>();
    }

    protected AlgorithmBase(final List<Rule> rules, final List<Aspect> aspects) {
        this.rules = rules;
        this.aspects = aspects;
    }

    @Override
    public void run() {
        runAlgorithm();
    }

    @Override
    public List<Rule> getOptimizedRules() {
        return optimizedRules;
    }

    @Override
    public List<Aspect> getAspects() {
        return aspects;
    }

    protected abstract void runAlgorithm();
}

```

\\src\\main\\java\\ru\\goncharov\\dissertation\\algorithms\\SingleIterationAlgorithm.java

```

package ru.goncharov.dissertation.algorithms;

import ru.goncharov.dissertation.algorithms.common.Aspect;
import ru.goncharov.dissertation.algorithms.common.AspectIndex;
import ru.goncharov.dissertation.algorithms.common.Rule;
import ru.goncharov.dissertation.algorithms.common.RuleLoaded;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

import static ru.goncharov.dissertation.algorithms.common.Aspect.ASPECT_BASE_ID;

public class SingleIterationAlgorithm extends AlgorithmBase {

    public static final int DEPTH = 3; // depth of analysis

    public SingleIterationAlgorithm(final RuleLoaded ruleLoaded) {
        super(ruleLoaded);
    }

    public SingleIterationAlgorithm(final List<Rule> rules, final List<Aspect>
aspects) {
        super(rules, aspects);
    }

    @Override
    protected void runAlgorithm() {
        for (int i = 0; i < rules.size(); i++) {
            final Rule rule = rules.get(i);
            final List<String> antecedent = rule.getAntecedent();
            if (antecedent.size() >= DEPTH) {
                final List<String> antecedentSubList = antecedent.subList(0,
DEPTH);
                final List<Rule> containsAntecedent =

```

```

        rules.stream()
            .filter(r ->
r.getAntecedent().containsAll(antecedentSubList))
            .collect(Collectors.toList());
        if (containsAntecedent.size() > 1) { // exist aspect
            addToOptimizedRule(
                containsAntecedent,
                createNewAspect(antecedentSubList)
            );
        } else {
            optimizedRules.add(rule);
        }
    } else {
        optimizedRules.add(rule);
    }
}
}

private Aspect createNewAspect(final List<String> antecedentSubList) {
    final Aspect aspect;
    final Optional<Aspect> any =
        aspects.stream()
            .filter(a -> a.getAntecedent().equals(antecedentSubList))
            .findAny();
    if (any.isPresent()) {
        aspect = any.get();
    } else {
        aspect = new Aspect(ASPECT_BASE_ID + new AspectIndex(aspects).value(),
antecedentSubList);
        aspects.add(aspect);
    }
    return aspect;
}

private void addToOptimizedRule(final List<Rule> containsAntecedent, final
Aspect aspect) {
    containsAntecedent.forEach(rule -> {
        final Rule newRule = createNewRule(aspect, rule);
        if (!optimizedRules.contains(newRule)) {
            optimizedRules.add(newRule);
        }
    });
}

private Rule createNewRule(final Aspect aspect, final Rule rule) {
    final List<String> newAntecedent = new ArrayList<>();
    newAntecedent.add(aspect.getId());
    newAntecedent.addAll(rule.getAntecedent().subList(DEPTH,
rule.getAntecedent().size()));
    return new Rule(newAntecedent, rule.getConsequent());
}
}
}

```

`\src\main\java\ru\goncharov\dissertation\algorithms\RecursiveAlgorithm.java`

```

package ru.goncharov.dissertation.algorithms;

import ru.goncharov.dissertation.algorithms.common.Aspect;
import ru.goncharov.dissertation.algorithms.common.AspectExist;
import ru.goncharov.dissertation.algorithms.common.Rule;
import ru.goncharov.dissertation.algorithms.common.RuleLoaded;

import java.util.List;

```

```

public class RecursiveAlgorithm extends AlgorithmBase {

    private final RuleLoaded ruleLoaded; // source set
    private Algorithm algorithm;

    public RecursiveAlgorithm(final RuleLoaded ruleLoaded) {
        this.ruleLoaded = ruleLoaded;
    }

    @Override
    protected void runAlgorithm() {
        algorithm = new SingleIterationAlgorithm(ruleLoaded);
        algorithm.run();
        while (new AspectExist(algorithm.getOptimizedRules()).value()) { //
recursive
            algorithm = new
SingleIterationAlgorithm(algorithm.getOptimizedRules(), algorithm.getAspects());
            algorithm.run();
        }

        @Override
        public List<Rule> getOptimizedRules() {
            return algorithm.getOptimizedRules();
        }

        @Override
        public List<Aspect> getAspects() {
            return algorithm.getAspects();
        }
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\Aspect.java

```

package ru.goncharov.dissertation.algorithms.common;

import java.util.List;

public class Aspect {

    public static final String ASPECT_BASE_ID = "aspect";
    private String id;
    private List<String> antecedent;

    public Aspect(String id, List<String> antecedent) {
        this.id = id;
        this.antecedent = antecedent;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public List<String> getAntecedent() {
        return antecedent;
    }

    public void setAntecedent(List<String> antecedent) {
        this.antecedent = antecedent;
    }
}

```

```

    @Override
    public String toString() {
        return String.format("%s=[%s %s %s];", id, antecedent.get(0),
            antecedent.get(1), antecedent.get(2));
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\AspectExist.java

```

package ru.goncharov.dissertation.algorithms.common;

import java.util.List;
import java.util.stream.Collectors;

import static ru.goncharov.dissertation.algorithms.SingleIterationAlgorithm.DEPTH;

public class AspectExist {

    private final List<Rule> rules;

    public AspectExist(final List<Rule> rules) {
        this.rules = rules;
    }

    public boolean value() {
        boolean result = false;
        for (int i = 0; i < rules.size(); i++) {
            final Rule rule = rules.get(i);
            final List<String> antecedent = rule.getAntecedent();
            if (antecedent.size() >= DEPTH) {
                final List<String> antecedentSubList = antecedent.subList(0,
                    DEPTH);

                final List<Rule> containsAntecedentSubList =
                    rules.stream()
                        .filter(r ->
                            r.getAntecedent().containsAll(antecedentSubList))
                        .collect(Collectors.toList());

                if (containsAntecedentSubList.size() > 1) { // exist aspect
                    result = true;
                    break;
                }
            }
        }
        return result;
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\AspectIndex.java

```

package ru.goncharov.dissertation.algorithms.common;

import java.util.Comparator;
import java.util.List;

import static ru.goncharov.dissertation.algorithms.common.Aspect.ASPECT_BASE_ID;

public class AspectIndex {

    private final List<Aspect> aspects;

    public AspectIndex(final List<Aspect> aspects) {
        this.aspects = aspects;
    }
}

```

```

    public int value() {
        int result;
        if (aspects.isEmpty()) {
            result = 1;
        } else {
            result = aspects.stream()
                .map(aspect ->
                    Integer.parseInt(aspect.getId().replace(ASPECT_BASE_ID, "")))
                .max(Comparator.comparing(Integer::valueOf))
                .get() + 1;
        }
        return result;
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\Rule.java

```

package ru.goncharov.dissertation.algorithms.common;

import com.google.common.base.Objects;

import java.util.List;

public class Rule {
    public final static String IF = "ECJM";
    private List<String> antecedent;
    private List<String> consequent;
    public final static String THEN = "TO";

    public Rule() {
    }

    public Rule(List<String> antecedent, List<String> consequent) {
        if (antecedent.contains(IF)) {
            antecedent.remove(IF);
        }
        if (consequent.contains(THEN)) {
            consequent.remove(THEN);
        }
        this.antecedent = antecedent;
        this.consequent = consequent;
    }

    public List<String> getAntecedent() {
        return antecedent;
    }

    public void setAntecedent(List<String> antecedent) {
        this.antecedent = antecedent;
    }

    public List<String> getConsequent() {
        return consequent;
    }

    public void setConsequent(List<String> consequent) {
        this.consequent = consequent;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Rule rule = (Rule) o;
    }
}

```

```

        return Objects.equal(antecedent, rule.antecedent) &&
            Objects.equal(consequent, rule.consequent);
    }

    @Override
    public int hashCode() {
        return Objects.hashCode(antecedent, consequent);
    }

    @Override
    public String toString() {
        return String.format(
            "%s %s %s %s;",
            IF,
            new RuleVisualized(antecedent).value(),
            THEN,
            new RuleVisualized(consequent).value()
        );
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleAntecedents.java

```

package ru.goncharov.dissertation.algorithms.common;

import java.util.ArrayList;
import java.util.List;

public class RuleAntecedents {

    private final String[] ruleFragments;

    public RuleAntecedents(final String[] ruleFragments) {
        this.ruleFragments = ruleFragments;
    }

    public List<String> value() {
        final List<String> antecedent = new ArrayList<>();
        for (int i = 0; i < ruleFragments.length; i++) {
            final String ruleFragment = ruleFragments[i].trim();
            if (!ruleFragment.equals(Rule.THEN)) {
                antecedent.add(ruleFragment);
            } else {
                break;
            }
        }
        return antecedent;
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleConsequents.java

```

package ru.goncharov.dissertation.algorithms.common;

import java.util.ArrayList;
import java.util.List;

public class RuleConsequents {

    private final String[] ruleFragments;

    public RuleConsequents(final String[] ruleFragments) {
        this.ruleFragments = ruleFragments;
    }
}

```

```

public List<String> value() {
    final List<String> consequent = new ArrayList<>();
    for (int i = ruleFragments.length - 1; i >= 0; i--) {
        final String ruleFragment = ruleFragments[i].trim();
        if (!ruleFragment.equals(Rule.THEN)) {
            consequent.add(ruleFragment);
        } else {
            break;
        }
    }
    return consequent;
}
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleFragments.java

```

package ru.goncharov.dissertation.algorithms.common;

```

```

public class RuleFragments {

    private final String rule;

    public RuleFragments(final String rule) {
        this.rule = rule;
    }

    public String[] value() {
        return rule
            .trim()
            .replace(";", "")
            .split(" ");
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleLoaded.java

```

package ru.goncharov.dissertation.algorithms.common;

```

```

import java.util.List;

```

```

public interface RuleLoaded {
    List<Rule> data();
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleLoadedBase.java

```

package ru.goncharov.dissertation.algorithms.common;

```

```

public abstract class RuleLoadedBase implements RuleLoaded {

    protected final String pathToResource;

    public RuleLoadedBase(final String pathToResource) {
        this.pathToResource = pathToResource;
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleLoadedFromFile.java

```

package ru.goncharov.dissertation.algorithms.common;

```

```

import com.google.common.base.Charsets;
import com.google.common.io.Files;

```

```

import java.io.File;
import java.io.IOException;
import java.util.List;

```

```

import java.util.concurrent.CopyOnWriteArrayList;

public class RuleLoadedFromFile extends RuleLoadedBase {

    public RuleLoadedFromFile(final String pathToResource) {
        super(pathToResource);
    }

    @Override
    public List<Rule> data() {
        List<Rule> rules = new CopyOnWriteArrayList<>();
        try {
            System.err.println(String.format("read file %s.", pathToResource));
            Files.readLines(
                new File(pathToResource),
                Charsets.UTF_8
            ).forEach(rule -> {
                final String[] ruleFragments = new RuleFragments(rule).value();
                rules.add(
                    new Rule(
                        new RuleAntecedents(ruleFragments).value(),
                        new RuleConsequents(ruleFragments).value()
                    )
                );
            });
        } catch (IOException e) {
            e.printStackTrace();
        }
        return rules;
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleLoadedFromJson.java

```

package ru.goncharov.dissertation.algorithms.common;

import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;
import org.json.simple.JSONArray;
import org.json.simple.parser.JSONParser;

import java.io.FileReader;
import java.util.List;
import java.util.concurrent.CopyOnWriteArrayList;

public class RuleLoadedFromJson extends RuleLoadedBase {

    public RuleLoadedFromJson(final String pathToResource) {
        super(pathToResource);
    }

    @Override
    public List<Rule> data() {
        List<Rule> rules = new CopyOnWriteArrayList<>();
        final JSONParser jsonParser = new JSONParser();
        try {
            System.err.println(String.format("read file %s.", pathToResource));
            final JSONArray jsonArray = (JSONArray) jsonParser.parse(new
FileReader(pathToResource));
            rules = new ObjectMapper()
                .readValue(
                    jsonArray.toString(),
                    new TypeReference<CopyOnWriteArrayList<Rule>>() {

```

```

        );
    } catch (Exception e) {
        e.printStackTrace();
    }
    return rules;
}
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleSaved.java

```

package ru.goncharov.dissertation.algorithms.common;

public interface RuleSaved {
    void create();
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleSavedBase.java

```

package ru.goncharov.dissertation.algorithms.common;

import ru.goncharov.dissertation.algorithms.Algorithm;

public abstract class RuleSavedBase implements RuleSaved {

    protected final Algorithm algorithm;
    protected final String outputFile;

    public RuleSavedBase(final Algorithm algorithm, final String outputFile) {
        this.algorithm = algorithm;
        this.outputFile = outputFile;
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleSavedToFile.java

```

package ru.goncharov.dissertation.algorithms.common;

import ru.goncharov.dissertation.algorithms.Algorithm;

import java.io.FileWriter;
import java.io.IOException;

public class RuleSavedToFile extends RuleSavedBase {

    public RuleSavedToFile(final Algorithm algorithm, final String outputFile) {
        super(algorithm, outputFile + ".txt");
    }

    @Override
    public void create() {
        try (final FileWriter writer = new FileWriter(outputFile)) {
            System.err.println(String.format("created file %s.", outputFile));
            for (final Rule rule : algorithm.getOptimizedRules()) {
                writer.write(rule + "\n");
            }
            writer.write("\n");
            for (final Aspect aspect : algorithm.getAspects()) {
                writer.write(aspect + "\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleSavedToJson.java

```

package ru.goncharov.dissertation.algorithms.common;

import com.fasterxml.jackson.databind.ObjectMapper;
import ru.goncharov.dissertation.algorithms.Algorithm;

import java.io.FileWriter;
import java.io.IOException;

public class RuleSavedToJson extends RuleSavedBase {

    public RuleSavedToJson(final Algorithm algorithm, final String outputFile) {
        super(algorithm, outputFile + ".json");
    }

    @Override
    public void create() {
        final ObjectMapper objectMapper = new ObjectMapper();
        final String rulesOutputFile = "rules_" + outputFile;
        try (final FileWriter writer = new FileWriter(rulesOutputFile)) {
            System.err.println(String.format("created file %s.",
rulesOutputFile));
writer.write(objectMapper.writeValueAsString(algorithm.getOptimizedRules()));
        } catch (IOException e) {
            e.printStackTrace();
        }
        final String aspectsOutputFile = "aspects_" + outputFile;
        try (final FileWriter writer = new FileWriter(aspectsOutputFile)) {
            System.err.println(String.format("created file %s.",
aspectsOutputFile));
            writer.write(objectMapper.writeValueAsString(algorithm.getAspects()));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

\src\main\java\ru\goncharov\dissertation\algorithms\common\RuleVisualized.java

```

package ru.goncharov.dissertation.algorithms.common;

import com.google.common.base.Joiner;

import java.util.List;

public class RuleVisualized {

    private final List<String> list;

    public RuleVisualized(final List<String> list) {
        this.list = list;
    }

    public String value() {
        final StringBuilder builder = new StringBuilder();
        if (list.size() == 1) {
            builder.append(list.get(0));
        } else {
            return Joiner.on(' ').join(list);
        }
        return builder.toString();
    }
}

```

## \pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>ru.goncharov.dissertation.demo</groupId>
  <artifactId>dissertation-demo</artifactId>
  <version>1.0</version>

  <packaging>jar</packaging>

  <properties>
    <jdk.version>11</jdk.version>
    <projectEncoding>UTF-8</projectEncoding>

<resourceDirectory>${project.basedir}/src/main/resources</resourceDirectory>
    <libDirectory>${project.build.directory}/lib</libDirectory>
    <outputDirectory>${project.build.outputDirectory}</outputDirectory>

    <guava.version>27.1-jre</guava.version>
    <jackson.version>2.9.8</jackson.version>
    <json-simple.version>1.1.1</json-simple.version>
    <commons-lang3.version>3.9</commons-lang3.version>
    <commons-io.version>2.6</commons-io.version>

    <junit.version>5.4.2</junit.version>

    <maven-resources-plugin.version>2.7</maven-resources-plugin.version>
    <maven-compiler-plugin.version>3.5.1</maven-compiler-plugin.version>
    <maven-jar-plugin.version>2.6</maven-jar-plugin.version>
    <maven-dependency-plugin.version>2.10</maven-dependency-
plugin.version>
    <maven-assembly-plugin.version>2.6</maven-assembly-plugin.version>
    <maven-surefire-plugin.version>2.22.1</maven-surefire-
plugin.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>com.google.guava</groupId>
      <artifactId>guava</artifactId>
      <version>${guava.version}</version>
    </dependency>
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>${jackson.version}</version>
    </dependency>
    <dependency>
      <groupId>com.googlecode.json-simple</groupId>
      <artifactId>json-simple</artifactId>
      <version>${json-simple.version}</version>
    </dependency>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
      <version>${commons-lang3.version}</version>
    </dependency>
  </dependencies>
</project>
```

```

<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>${commons-io.version}</version>
</dependency>

<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>${junit.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>${junit.version}</version>
  <scope>test</scope>
</dependency>
</dependencies>

<build>
  <finalName>DissertationAlgorithmsDemoApp-${version}</finalName>
  <sourceDirectory>src/main/java</sourceDirectory>
  <testSourceDirectory>src/test/java</testSourceDirectory>
  <resources>
    <resource>
      <directory>${resourceDirectory}</directory>
      <filtering>>false</filtering>
    </resource>
  </resources>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-resources-plugin</artifactId>
      <version>${maven-resources-plugin.version}</version>
      <configuration>
        <encoding>${projectEncoding}</encoding>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>${maven-compiler-plugin.version}</version>
      <configuration>
        <source>${jdk.version}</source>
        <target>${jdk.version}</target>
        <encoding>${projectEncoding}</encoding>
        <showWarnings>>true</showWarnings>
        <showDeprecation>>true</showDeprecation>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>${maven-jar-plugin.version}</version>
      <configuration>
        <archive>
          <index>>true</index>
          <addMavenDescriptor>>true</addMavenDescriptor>
          <compress>>true</compress>
          <manifest>
            <addClasspath>>true</addClasspath>
            <classpathPrefix>lib</classpathPrefix>

```

```

<addDefaultImplementationEntries>>true</addDefaultImplementationEntries>
<addDefaultSpecificationEntries>>true</addDefaultSpecificationEntries>
<mainClass>ru.goncharov.dissertation.DemoApp</mainClass>
  </manifest>
</archive>
</configuration>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>${maven-dependency-plugin.version}</version>
  <executions>
    <execution>
      <id>copy-dependencies</id>
      <phase>package</phase>
      <goals>
        <goal>copy-dependencies</goal>
      </goals>
      <configuration>
<outputDirectory>${libDirectory}</outputDirectory>
        <overwriteReleases>>false</overwriteReleases>
        <overwriteSnapshots>>true</overwriteSnapshots>
        <overwriteIfNewer>>true</overwriteIfNewer>
      </configuration>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-assembly-plugin</artifactId>
  <version>${maven-assembly-plugin.version}</version>
  <configuration>
    <descriptorRefs>
      <descriptorRef>jar-with-dependencies</descriptorRef>
    </descriptorRefs>
  <archive>
    <manifest>
<mainClass>ru.goncharov.dissertation.DemoApp</mainClass>
    </manifest>
  </archive>
  </configuration>
  <executions>
    <execution>
      <id>release</id>
      <phase>package</phase>
      <goals>
        <goal>single</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>${maven-surefire-plugin.version}</version>
</plugin>
</plugins>
</build>
</project>

```