

Государственное образовательное учреждение высшего профессионального образования



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация

КАФЕДРА Компьютерные системы автоматизации производства

Р А С Ч Ё Т Н О - П О Я С Н И Т Е Л Ь Н А Я З А П И С К А

к дипломному проекту на тему:

Интегрированная система поддержки принятия решения о выборе
режимов механической обработки тонкостенных деталей

Студент _____ П.А. Еремейкин
(Подпись, дата) (И.О.Фамилия)

Руководитель дипломного проекта _____ А.Д. Жаргалова
(Подпись, дата) (И.О.Фамилия)

Консультант по
научно-исследовательской части _____ _____
(Подпись, дата) (И.О.Фамилия)

Консультант по конструкторской части _____ _____
(Подпись, дата) (И.О.Фамилия)

Консультант по технологической части _____ _____
(Подпись, дата) (И.О.Фамилия)

Консультант по
организационно-экономической части _____ Л.А. Силаева
(Подпись, дата) (И.О.Фамилия)

Консультант по охране труда и экологии _____ А.С. Козодаев
(Подпись, дата) (И.О.Фамилия)

СОДЕРЖАНИЕ

1 Введение	6
2 Техническое задание	10
2.1 Основания для разработки	10
2.2 Назначение разработки	10
2.3 Требования к функциональным характеристикам	10
2.4 Требования к надежности	11
2.5 Условия эксплуатации	11
2.6 Требования к составу и параметрам технических средств . . .	11
2.7 Требования к информационной и программной совместимости	12
2.8 Требования к маркировке и упаковке	12
2.9 Требования к транспортированию и хранению	12
2.10 Требования к программной документации	12
2.11 Технико-экономические показатели	12
2.12 Стадии и этапы разработки	12
2.13 Порядок контроля и приемки	13
2.14 Приложения	13
2.15 Предпроектное исследование	13
3 Разработка концепции автоматизированной системы	14
3.1 Принцип моделирования	14
3.2 Принцип модульности	16
3.3 Принцип комплексности	17
3.4 Принцип независимости	18
4 Предпроектное исследование	19
4.1 Формулировка целей	19
4.2 Выбор языка программирования	20
4.3 Выбор модульной платформы	22
4.4 Выбор системы моделирования	23

5 Эскизное проектирование	25
5.1 Определение функционала системы	25
5.2 Разработка структуры входных и выходных данных	27
5.2.1 Входные данные для анализа деформаций	27
5.2.2 Входные данные для калькулятора режимов резания .	30
5.3 Эскизное описание принципов работы	31
5.4 Модули	32
5.4.1 Концепции NetBeans	32
5.4.2 Применение платформы	33
6 Техническое проектирование	35
6.1 Уточнение структуры входных данных	35
6.2 Чтение входных данных	39
6.2.1 Подключение базы данных	39
6.2.2 Выбор СУБД	39
6.2.3 Реализация шаблона DAO	40
6.3 Построение МКЭ модели	43
6.3.1 Выбор типа конечных элементов	43
6.3.2 Допущения	45
6.3.3 Параметры модели	45
6.3.4 Модель в Abaqus	47
6.4 Разработка калькулятора резания	48
6.4.1 Библиотека Drools	49
6.4.2 Запись продукционных правил	50
7 Расчет затрат на разработку интегрированной системы поддержки принятия решения	54
7.1 Введение	54
7.2 Организация и планирование процесса разработки	54
7.3 Расчет трудоемкости этапов	56
7.3.1 Расчет трудоемкости разработки технического задания	56
7.3.2 Расчет трудоемкости выполнения эскизного проекта .	57

7.3.3	Расчет трудоемкости выполнения технического проекта	57
7.3.4	Расчет трудоемкости выполнения рабочего проекта	58
7.3.5	Расчет трудоемкости стадии внедрения	60
7.3.6	Расчет суммарной трудоемкости и продолжительности работ	60
7.4	Определение затрат на разработку интегрированной системы	61
7.4.1	Расчет основной заработной платы	61
7.4.2	Расчет дополнительной заработной платы	62
7.4.3	Отчисления на социальное страхование	62
7.4.4	Накладные расходы	63
7.4.5	Амортизационные отчисления на оборудование	63
7.4.6	Результаты расчета затрат	64
7.5	Выводы	64
8	Охрана труда и экология	65
8.1	Введение	65
8.2	Опасные факторы	65
8.2.1	Воздействие электрического тока	66
8.2.2	Взрыв кинескопа	66
8.2.3	Возгорание устройств	67
8.3	Вредные факторы	68
8.3.1	Повышенный уровень шума	69
8.3.2	Повышенная запыленность воздуха	70
8.3.3	Недостаточная или избыточная освещенность рабочей зоны	70
8.3.4	Повышенный уровень электромагнитного поля	71
8.3.5	Умственное напряжение	72
8.3.6	Напряжение зрительных анализаторов	72
8.3.7	Монотонность труда	73
8.3.8	Эмоциональные нагрузки	73
8.4	Результаты анализа	74
8.5	Расчет заземления	74

8.5.1	Определение предельно допустимого сопротивления заземляющего устройства	75
8.5.2	Определение сопротивления естественных заземителей	76
8.5.3	Определение сопротивления искусственных заземителей	77
8.6	Утилизация печатных плат	80
8.6.1	Технологии переработки печатных плат	80
8.6.2	Физическая переработка	81
8.6.3	Химическая переработка	82
8.6.4	Использование неметаллических фракций	83
8.6.5	Выводы	84
	Список литературы	86

ГЛАВА 1

ВВЕДЕНИЕ

Эксплуатационные характеристики изделий машиностроения в большой мере зависят от качества изготовления деталей, составляющих изделие. Качество изготовления определяется степенью соответствия параметров готовой детали и параметров, достижение которых требуется документацией. В роли такого параметра может выступать точность изготовления геометрических размеров, определяемая допуском.

Погрешности различных видов препятствуют абсолютно точному воспроизведению заданных параметров. Например, на точность изготовления деталей влияют погрешности базирования, измерения, настройки инструмента и т.д. При обработке нежестких деталей на первый план выходят погрешности, связанные с деформацией самой детали.

Рассмотрим пример обработки тонкостенной детали в трехкулачковом патроне токарного станка. Схема обработки изображена на рисунке 12. Цифровыми сносками обозначены: 1 - кулачок патрона, 2 - цилиндрическая заготовка, 3 - токарный резец. До начала обработки деталь находится в недеформированном состоянии (рисунок 1а). После закрепления детали силы, действующие со стороны кулачков, деформируют заготовку таким образом, что профиль заготовки отличается от идеально цилиндрического (рисунок 1б). Дополнительно заготовка деформируется под действием силы резания, а также в связи с нагревом. В процессе резания инструмент неравномерно снимает припуск по окружности заготовки (рисунок 1в). После окончания обработки и снятия усилий закрепления отклонение от круглости заготовки достигает величин, сравнимых с величиной допуска и в зависимости от конкретных условий может превышать его (рисунок 1г).

На практике в случае обработки нежестких деталей применяется специальное технологическое оборудование, позволяющее уменьшить возникающие деформации. Использование сырых кулачков, растрачиваемых под диаметр заготовки - один из способов уменьшения деформаций закрепления. После растигивания кулачок охватывает деталь по большей площади.

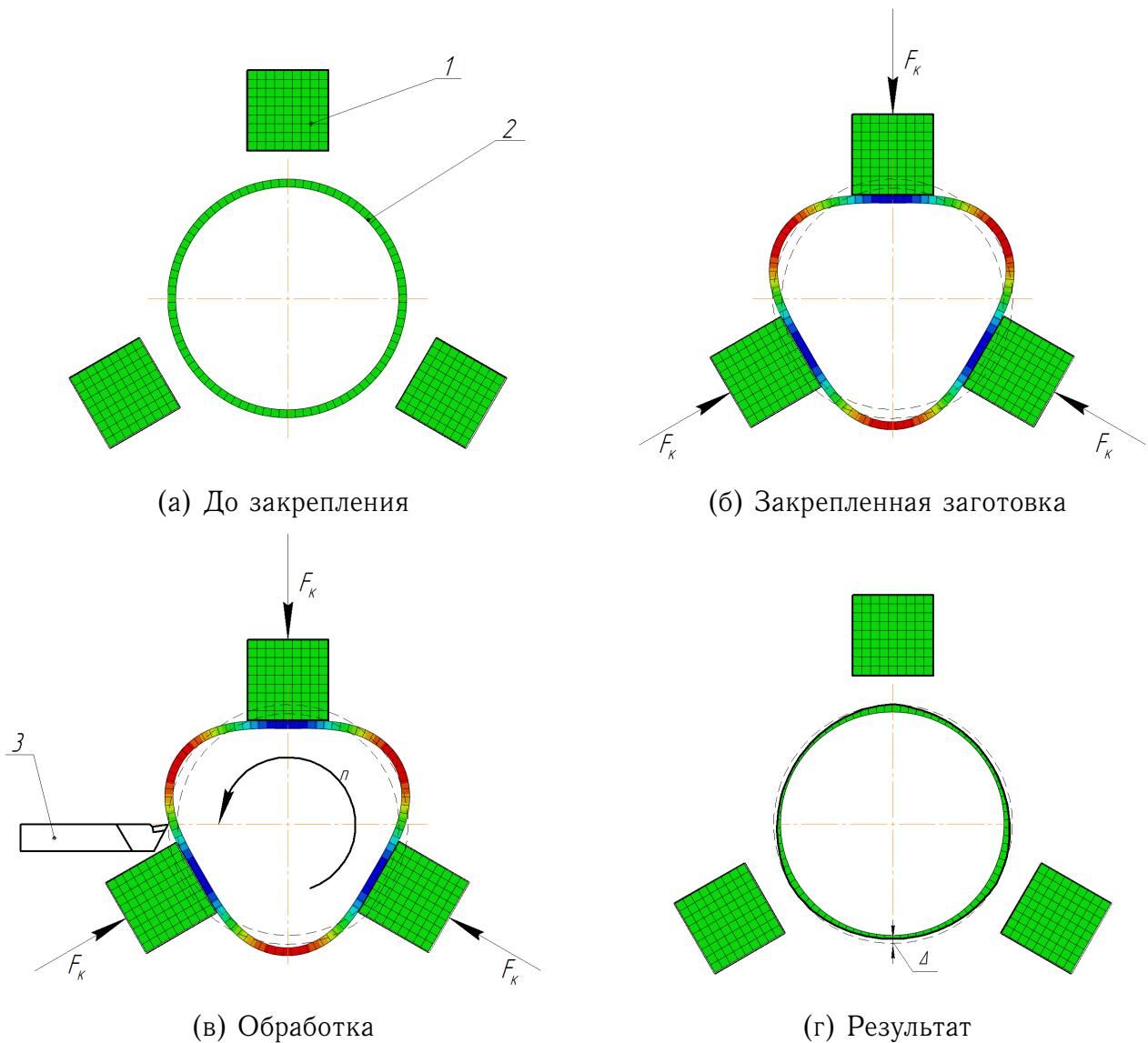


Рисунок 1 – Обработка тонкостенной заготовки

Также используется обработка с технологическим заполнителем (например, легкоплавким материалом), который увеличивает жесткость детали на время обработки и выплавляется после окончания обработки. Вариант обработки на нормативных режимах резания в специальных приспособлениях привлекателен тем, что он позволяет применять апробированные технологические процессы. Однако, необходимость создания при этом специальных приспособлений требует значительных материальных, производственных и временных затрат. Это удорожает технологическую подготовку производства и увеличивает её сроки.

Ввиду указанных ограничений на существующие способы обработки, предлагается рассмотреть еще один метод - подбор режимов обработки и условий закрепления таким образом, чтобы обеспечить изготовление размеров в рамках допуска. Предполагается, что такой метод может найти применение прежде всего в единичном и мелкосерийном производстве, располагающим, как правило, только универсальным оборудованием. Суть подхода заключается в том, чтобы заранее, на этапе разработки технологической документации, определить режимы резания, менее эффективные с точки зрения производительности, но оптимальные с точки зрения точности изготовления. С уменьшением сил резания уменьшаются и деформации заготовки, а также требуемые усилия закрепления, но с другой стороны, увеличивается время обработки. Подобрав баланс между скоростью и точностью обработки, технолог имеет возможность назначить режимы резания, приемлемые для изготовления заданной детали без привлечения дополнительной технологической оснастки. Достоинством такого подхода является меньшая ресурсоёмкость и продолжительность технологической подготовки производства. Однако, данный подход в настоящее время ещё не получил достаточного научного обоснования и, как следствие, не поддержан методическими рекомендациями, необходимыми для его применения в промышленных масштабах [1].

Для поддержки решения о назначении режимов резания предлагается разработать информационную систему, позволяющую анализировать деформации заготовки при заданных геометрических параметрах и режимах резания. Такой инструмент можно использовать для последовательной проверки ряда значений параметров процесса и выбора наиболее рациональных.

Система, спроектированная в соответствии с современными тенденциями в разработке ПО станет эффективным инструментом для назначения режимов обработки. Объектно-ориентированная модульная архитектура такой системы позволит расширять её функционал в случае возникновения новых требований к применению программы, а независимость от конкрет-

ной операционной системы позволит легко внедрить такие системы в процесс разработки технологической документации на предприятиях безотносительно конкретной существующей информационной среды.

ГЛАВА 2

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

2.1 Основания для разработки

Основанием для разработки системы являются документы:

1. Задание на выполнение дипломного проекта
2. Календарный план на выполнение дипломного проекта

2.2 Назначение разработки

Функциональным назначением разработки является обеспечение интегрированной рабочей среды для гибкого моделирования процессов деформирования тонкостенных заготовок в процессе их токарной обработки. Под гибкостью понимается возможность изменения параметров процесса в зависимости от расчетного случая и конфигурации заготовки. Эксплуатационным назначением является обеспечение инструментального средства для последовательного определения рациональных режимов резания тонкостенных заготовок и автоматизация сопутствующих расчетов.

2.3 Требования к функциональным характеристикам

Система информационной поддержки должна обеспечивать выполнение следующих функций:

1. Расчет деформаций тонкостенной заготовки при токарной обработке с закреплением в кулачковом патроне.
2. Автоматизация необходимых сопутствующих расчетов (режимов резания)

3. Изменение параметров рассматриваемого процесса: геометрии заготовки и оснастки и значений силовых факторов (силы резания и закрепления)
4. Представление результатов расчета в виде графиков или таблиц деформаций
5. Генерация отчета по результатам расчета
6. Вывод дополнительных информационных отладочных сообщений

2.4 Требования к надежности

Разрабатываемая система должна обеспечивать надежную работу в условиях ошибочного пользовательского ввода. При возникновении системных исключительных ситуаций необходимо обеспечить их обработку чтобы не допустить аварийного завершения программы.

2.5 Условия эксплуатации

Аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью $220\text{V} \pm 10\%$, 50 Гц с защитным заземлением при следующих климатических условиях:

- температура окружающей среды – от 15 до 30 градусов С;
- относительная влажность воздуха – от 30% до 80%;
- атмосферное давление – от 630 мм. р.с. до 800 мм. р.с.

2.6 Требования к составу и параметрам технических средств

Программный продукт должен работать на компьютерах со следующими характеристиками:

- объем ОЗУ не менее 2 Гб;
- объем жёсткого диска не менее 40 Гб;
- микропроцессор с тактовой частотой не менее 1.5 ГГц;
- монитор с разрешением от 1024*768 и выше.

2.7 Требования к информационной и программной совместимости

Windows, Linux, и т.д.

2.8 Требования к маркировке и упаковке

Не предъявляются

2.9 Требования к транспортированию и хранению

Не предъявляются

2.10 Требования к программной документации

Не предъявляются

2.11 Технико-экономические показатели

? ? ?

2.12 Стадии и этапы разработки

? ? ?

2.13 Порядок контроля и приемки

???

2.14 Приложения

???

2.15 Предпроектное исследование

???

ГЛАВА 3

РАЗРАБОТКА КОНЦЕПЦИИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

3.1 Принцип моделирования

Согласно техническому заданию система должна обеспечить расчет деформаций тонкостенной заготовки в процессе токарной обработки. Токарная обработка характеризуется большим разнообразием возможных вариантов осуществления в зависимости от обрабатываемой поверхности и способа закрепления. Разработать и запрограммировать адекватную модель, основанную исключительно на математическом представлении известных теоретических закономерностях сопротивления материалов для каждого из возможных вариантов - чрезвычайно сложная задача с технической точки зрения.

Известно, что современные САЕ системы позволяют успешно решать задачи определения деформаций и напряжений для объектов практически неограниченной сложности благодаря использованию метода конечных элементов (МКЭ). После разбиения рассматриваемых объектов на конечные элементы определенного типа и наложении ограничений на эти элементы согласно закономерностям предметной области задача сводится к решению системы из большого количества уравнений. Успех решения полученной системы уравнений как правило определяется только отведенным на решение временем работы ЭВМ.

Тем не менее, сама по себе реализация метода конечных элементов в разрабатываемом продукте не возможна из-за большой трудоемкости и требует огромной исследовательской и проектной работы целого коллектива профессионалов. Таким образом, наилучшим выходом из сложившейся ситуации представляется использование готовой САЕ системы. Применение системы МКЭ расчета возможно, если она отвечает следующим трем основным требованиям:

1. Предоставляет возможность расчета заранее составленной модели под

управлением внешнего процесса операционной системы (в роли которого будет выступать разрабатываемая система анализа деформаций)

2. Имеет механизмы параметризации модели. В наилучшем случае параметризация должна достигаться благодаря выполнению пользовательских сценариев.
3. Позволяет получить результат расчета в виде графиков, диаграмм напряжений и деформаций а также в числовом виде для каждого элемента модели.

В рамках перечисленных соображений можно сформулировать концепцию системы следующим образом. На первом этапе система собирает информацию о значениях параметров модели через графический пользовательский интерфейс. По завершению ввода функция системы состоит в формировании задания для используемой готовой САЕ системы. На этапе подготовки задания производится дополнительный расчет необходимых величин, если они не были непосредственно указаны на первом этапе и могут быть получены расчетным путем. Третий этап состоит в вызове расчетного ядра САЕ системы по сформированному заданию. По завершению обработки модели разрабатываемая система агрегирует результаты и отображает их пользователю, при необходимости генерируя отчет.

Таким образом, основная идея состоит в расчете модели обработки с использованием сторонней САЕ системы. Предполагается, что роль разработчика моделей будет выполнять эксперт, как правило не являющийся пользователем системы. В задачи эксперта входит описание нового расчетного случая при помощи инструментов, предоставляемых конкретной выбранной системой МКЭ расчета и другие необходимые работы по подготовке системы. Готовая модель предоставляется в распоряжение пользователю, в задачи которого входит выбор наиболее подходящей модели из множества имеющихся и её загрузка в разрабатываемую систему поддержки.

3.2 Принцип модульности

Уже на этапе формулирования концепции системы становится видна её большая сложность. Для того чтобы сохранить контроль над расширяющейся по мере разработке системы и не допустить её деградации следует определиться с принимаемыми для этого мерами и также включить их в концепцию системы, так как принимаемые меры коренным образом повлияют на процесс проектирования.

В соответствии с накопленным мировым опытом в разработке программного обеспечения одним из наиболее жизнеспособных способов контроля сложности является разработка приложения согласно принципам объектно-ориентированного программирования. Во-первых, ООП методология позволяет добиться большого процента повторного использования кода благодаря принципам наследования и полиморфизма [2]. Во-вторых, принципы абстракции и инкапсуляции облегчают задачу программиста так как они существенно ограничивают область кода, который влияет на рассматриваемый участок программы и позволяют формулировать мысли в терминах предметной области.

Разрабатываемая система сложна не только по причине большой трудоемкости процесса разработки, но также и процесса поддержки готовой системы. В случае возникновения новых функциональных требований, не покрытых техническим заданием на первом этапе, потребуется совершить большой объем работ, если заранее не предусмотреть модульность системы. Модульность системы позволит “присоединить” к готовой системе недостающие функциональные элементы - модули, без переработки какой-либо существенной части самого приложения.

Как показано выше, сам по себе переход от структурного программирования к объектно-ориентированному - большой шаг вперед. Однако, такой шаг ещё не гарантирует структурированной модульной архитектуры приложения. Отдельный класс хоть и инкапсулирует некоторые данные, тем не менее не является модулем в смысле всего приложения. Для достиже-

ния этой цели потребуется применение специальных мер, зависящих от выбранного языка программирования. Такие меры могут представлять, например, использование определенного набора шаблонов проектирования, в случае реализации модульности системы своими силами, или использования одного из многочисленных фреймворков в другом случае.

3.3 Принцип комплексности

Принцип модульности оставляет широкие возможности для расширения функционала системы даже после выпуска готовой системы. Однако такой возможностью следует пользоваться в пределах разумного. Не следует исходить из этой концепции для оправдания отсутствия каких-либо востребованных функций. В контексте текущего проекта должен быть разработан исходный набор модулей, обеспечивающий выполнение большинства функций, которые могут понадобиться пользователю при работе в рамках рассматриваемой предметной области. Это означает что следует комплексно подойти к решению проблемы.

В рамках базового набора модулей согласно техническому заданию должны обязательно присутствовать модуль для расчета режимов резания, модуль графического отображения результатов и модуль генерации отчетов. Также следует включить в основной состав модули просмотра трехмерного изображения рассчитываемой модели и вывода текстовой информации, формируемой САЕ системой.

Кроме самого факта наличия модулей в системе необходимо обеспечить взаимный обмен информацией между ними. Например, передачу значений, полученных в модуле расчета режимов резания в модуль параметризации модели. Этот прием позволит избавить пользователя от случайных ошибок при копировании и сделает работу с системой более удобной. Согласно идеи комплексности, полезной особенностью может стать предоставление справочных данных для выбора в качестве значений вводимых параметров.

Таким образом, разрабатываемая система должна стать результатом

многосторонней проработки проблемы.

3.4 Принцип независимости

При разработке системы уже на самых ранних стадиях необходимо ориентироваться на возможное многообразие используемых на предприятиях программных и аппаратных средств. Известно, что для современного рынка информационных технологий характерна большая вариативность используемого программного обеспечения. В рамках проекта интерес представляют операционные системы и системы инженерных расчетов (САЕ). Для разных предприятий сочетания конкретных решений в этих областях могут быть различными, например, роль операционной системы может выполнять какая либо версия Microsoft Windows или Linux. В роли САЕ системы может использоваться Ansys, Abaqus, в некоторых случаях Siemens NX или SolidWorks.

В идеальном случае следует ориентироваться на независимость проектируемого приложения от индивидуальных особенностей какой-либо внешней системы. Это позволит расширить границы применения программы и облегчить задачу её освоения в рамках существующей информационной среды. Удовлетворить условиям концепции независимости поможет правильный выбор средств разработки а также определенные проектные меры, например, разработка межсистемного взаимодействия на основе соглашений (интерфейсов).

ГЛАВА 4

ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

4.1 Формулировка целей

Основной целью разработки является снижение трудоемкости назначения рациональных режимов резания. Эта цель может быть представлена в виде трех подцелей более низкого уровня:

1. Снижение трудоемкости предварительных расчетов
2. Снижение трудоемкости основных расчетов
3. Снижение трудоемкости анализа результатов

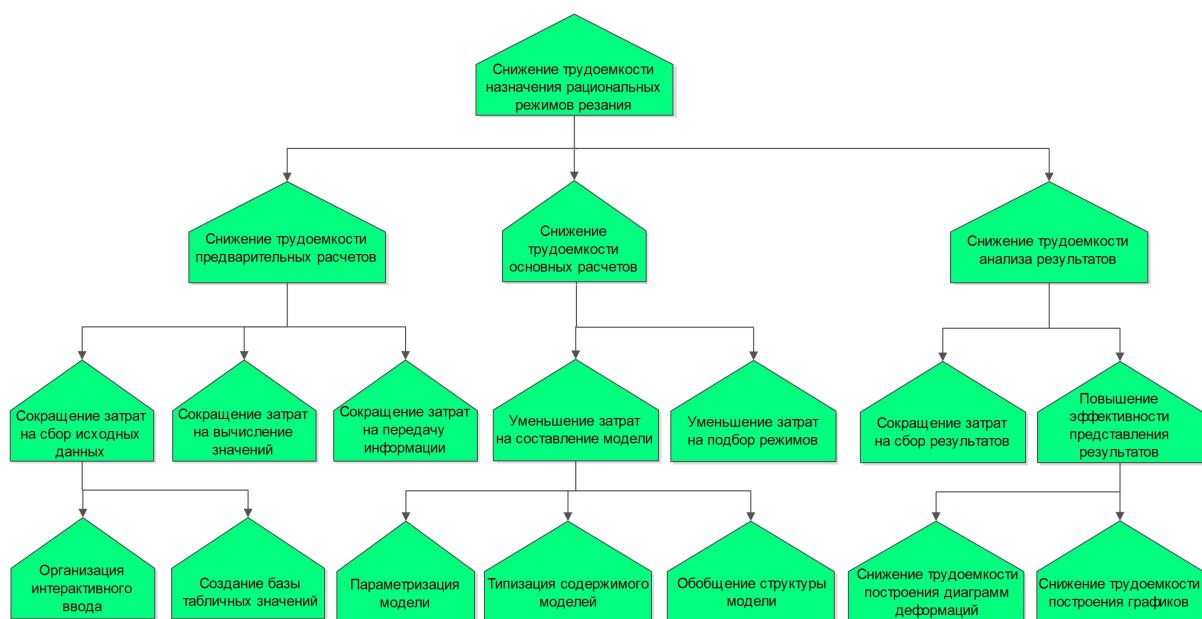


Рисунок 2 – Дерево целей

Снижение трудоемкости предварительных расчетов сводится к автоматизации вычисления режимов резания по традиционным методикам. Эта цель может быть достигнута при сокращении затрат на сбор исходных данных, вычисление значений и передачу информации. Под традиционными методиками понимается прежде всего таблично-аналитический подход,

описанный в [3]. Согласно указанной литературе, для вычисления параметров режима резания необходимо выбрать исходные данные из таблиц эмпирических коэффициентов и подставить их в приведенные в справочнике формулы. Снижение трудоемкости этих действий и образует первую подцель.

Вторая задача состоит в том, чтобы разработать общую структуру модели, а также типизировать её содержимое. Путем параметризации может быть достигнуто эффективное повторное применение одной модели в рамках разработки различных технологических процессов. Используя внутренний расчетный механизм системы планируется обеспечить уменьшение затрат на подбор рациональных режимов резания.

После проведения расчетов требуется сделать вывод о допустимости принятых параметров обработки. Автоматизация построения диаграмм и графиков деформаций позволит повысить эффективность анализа результатов.

Дерево целей изображено на рисунке 2. Оно систематизирует вышесказанное в виде иерархической структуры.

4.2 Выбор языка программирования

Выбору языка программирования следует уделить особое внимание. От правильного решения существенно зависит насколько трудоемким будет процесс написания программы. Существующие языки программирования сильно различаются по возможностям, гибкости и области применения. На данном этапе необходимо рассмотреть основные варианты и выявить наилучший.

На стадии разработки концепции было выявлено, что следует опираться на подходы объектно-ориентированного программирования. Этот вывод соответствующим образом сужает область рассматриваемых решений. Для повышения пригодности продукта к обслуживанию также ограничимся общеизвестными и широко распространенными языками. Дополнительная

выгода от использования таких языков состоит в возможности применения готовых компонентов и библиотек.

Под указанные ограничения подходят такие языки как C++, C#, Java, Python. Сравним эти языки по существенным факторам методом непосредственной оценки. Для оценки будут приняты во внимание следующие характеристики: распространенность, скорость разработки, гибкость, безопасность. В рамках проекта производительность не рассматривается так как она не является ограничивающим фактором. По каждому критерию языку будет присвоен балл от 1 до 10. Наиболее предпочтительным будет признан язык с наибольшим суммарным баллом.

Распространенность может быть определена по числу проектов на информационном ресурсе github.com и числу заданных вопросов на stackoverflow.com. Эти ресурсы выбраны в качестве референтных из-за большой популярности в IT сообществе. Аналитическая фирма RedMonk опубликовала объективный рейтинг языков программирования в соответствии с указанными критериями [4]. Этот отчет использован в качестве основания для присвоения баллов, поэтому Java получает максимальный балл, а остальные языки оценены меньшим числом.

Скорость разработки в большей мере субъективный параметр, так как он зависит не только от конструкций языка, но и от умений и опыта конкретного программиста, хотя можно выявить и общие тенденции. Скорость разработки рассматривается с учетом замечаний, изложенных в [5]. Известно, что разработка на C++ требует больших временных затрат ввиду сложности конструкций и большого количества деталей, которые должны быть учтены программистом. По этому критерию предпочтение отдано Python, который относят к языкам сценариев. C# и Java в плане скорости разработки не выделяются и получили средний балл.

Гибкость также не поддается количественному определению, однако она зависит только от самого языка. Динамическая типизация позволила Python набрать максимальный балл. Также высоко оценены возможности ручного управления памятью в C++.

Таблица 1 – Оценка языков программирования

Фактор	C++	C#	Java	Python
Распространенность	9	9	10	9
Скорость разработки	8	9	9	10
Гибкость	10	9	9	10
Безопасность	8	10	10	8
Итого	36	37	38	37

Безопасность в текущем контексте следует понимать как вероятность возникновения не обрабатываемых ошибок во время выполнения программы. Ошибки такого типа негативным образом сказываются на качестве программного продукта и наиболее затратны для исправления. Низкий балл Python обусловлен уже упомянутой особенностью - динамической типизацией. В данном случае программист не имеет возможности воспользоваться описанием ошибок компиляции для устранения несоответствия типов переменных, это приводит к не обрабатываемым исключительным ситуациям уже во время выполнения. C++ также получил более низкую оценку ввиду общей сложности разработки и возможных ошибок при использовании адресной арифметики.

Согласно таблице 1 наиболее рациональным выбором является язык Java. Следует отметить, что оценка произведена с учетом требований к конкретному проекту и для других случаев на первый план могут выйти другие факторы.

4.3 Выбор модульной платформы

При описании модульной архитектуры в рамках концепции был оставлен открытым вопрос о средствах реализации. После выбора языка программирования можно вернуться к этому вопросу и проработать его более детально.

Многие эксперты в области программирования рекомендуют не разра-

батывать сложные компоненты, если уже существуют достойные библиотеки, содержащие необходимый функционал [6]. В случае разработки на Java достичь модульной компоновки приложения можно при помощи одного из двух фреймворков: NetBeans Platform или Eclipse RCP. Детальный анализ показывает, что оба продукта в большей степени предоставляют эквивалентные возможности. Тем не менее были выделены следующие существенные различия: платформа NetBeans использует библиотеку Swing в качестве инструмента реализации пользовательского интерфейса, в то время как в Eclipse используется SWT. Немного более предпочтительным выглядит Swing, так как она является стандартной для Java и поставляется в комплекте разработчика (JDK - Java Development Kit) и поэтому имеет большое количество всевозможных сторонних расширений. Второе отличие в том, что в Eclipse модульная система реализована в соответствии со стандартом OSGi (Open Service Gateway Initiative) в то время как в NetBeans используется специфичная модульная система, хотя документация NetBeans также заявляет о частичной поддержке OSGi [7]. В рамках проекта это отличие не представляется существенным.

В качестве программной платформы было принято использовать NetBeans. На это решение повлияло наличие большого числа доступных ресурсов для освоения платформы, таких как проекты с открытым исходным кодом и обучающие официальные материалы а также подробная документация.

4.4 Выбор системы моделирования

Принцип моделирования подразумевает использование системы МКЭ расчетов при вычислении значений деформаций. Наиболее подходящими для этой цели были признаны ANSYS и SIMULA Abaqus FEA. Обе программы предоставляют богатый функционал, позволяющий моделировать рассматриваемый процесс обработки, поэтому функциональные критерии отходят на второй план. И ANSYS и Abaqus отвечают трем сформулиро-

ванным ранее требованиям к системе моделирования.

Для выдачи расчетного задания в случае использования ANSYS имеется возможность организации взаимодействия через интерфейс командной строки. Модель может быть составлена на специальном предметно-ориентированным языке ANSYS Parametric Design Language (APDL). Однако, анализ официальной документации не выявил ни одного приемлемого способа передачи фактических значений параметров модели при вызове выполнения задания. Эта проблема может быть решена путем непосредственной замены значений в текстовом файле задания, но такой путь в большей степени подвержен вероятному возникновению ошибок, менее гибкий и более трудоемкий. Экспорт геометрии из ANSYS возможен только в единственном формате IGES. Хотя IGES широко используется в коммерческих CAD системах, для него не существует бесплатных Java библиотек визуализации, а реализация такой библиотеки (с учетом размера спецификации IGES около 700 страниц) не предоставляется возможной.

В плане возможностей интеграции с внешними системами выгодно отличается Abaqus. В Abaqus для описания модели используется язык общего назначения Python. При вызове расчетного задания через интерфейс командной строки есть возможность передачи значений параметров (например, геометрических размеров или силовых факторов). Также Abaqus поддерживает открытый формат OBJ для обмена геометрией. OBJ — это простой формат данных, для него существуют Java библиотеки, позволяющие отображать сохраненные модели.

Таким образом, для моделирования процесса будет использован Abaqus так как эта система более пригодна для взаимодействия с внешней информационной средой.

ГЛАВА 5

ЭСКИЗНОЕ ПРОЕКТИРОВАНИЕ

5.1 Определение функционала системы

На данном этапе необходимо определить какие функции должна выполнять разрабатываемая система. Для графического представления будет применена UML модель вариантов использования. Она описывает не только состав функций, но и их взаимосвязь с множеством действующих лиц.

Основная идея диаграммы вариантов использования состоит в отображении требований к системе через две базовые сущности - действующие лица и варианты использования. Пользователи и другие системы, которые могут взаимодействовать с продуктом называются действующими лицами, а вариант использования представляет собой спецификацию поведения [8]. В системе предварительно можно выделить три лица, взаимодействующих с системой.

Первое лицо представлено пользователем, который заинтересован в выполнении функций, связанных с получением значений деформаций заготовки и их анализом. Для этого пользователю должна быть предоставлена возможность загрузки описания расчетного случая (то есть модели) в систему. Перед запуском расчета необходимо применить заданные характеристики процесса к параметризованной модели. По результатам работы САЕ системы должны быть получены графики и диаграммы деформаций. Также пользователь ожидает от системы выполнения вспомогательных функций, таких как расчет режимов резания.

Необходимость выделения такого лица как инженер по знаниям обусловлена результатами анализа литературы [3], который показал, что для расчета режимов резания по таблично-аналитической методике потребуется представление экспертных знаний в формальном виде. Решение этой задачи возлагается на инженера по знаниям, который должен иметь возможность корректировать содержимое базы знаний, первоначально составленной при разработке системы.

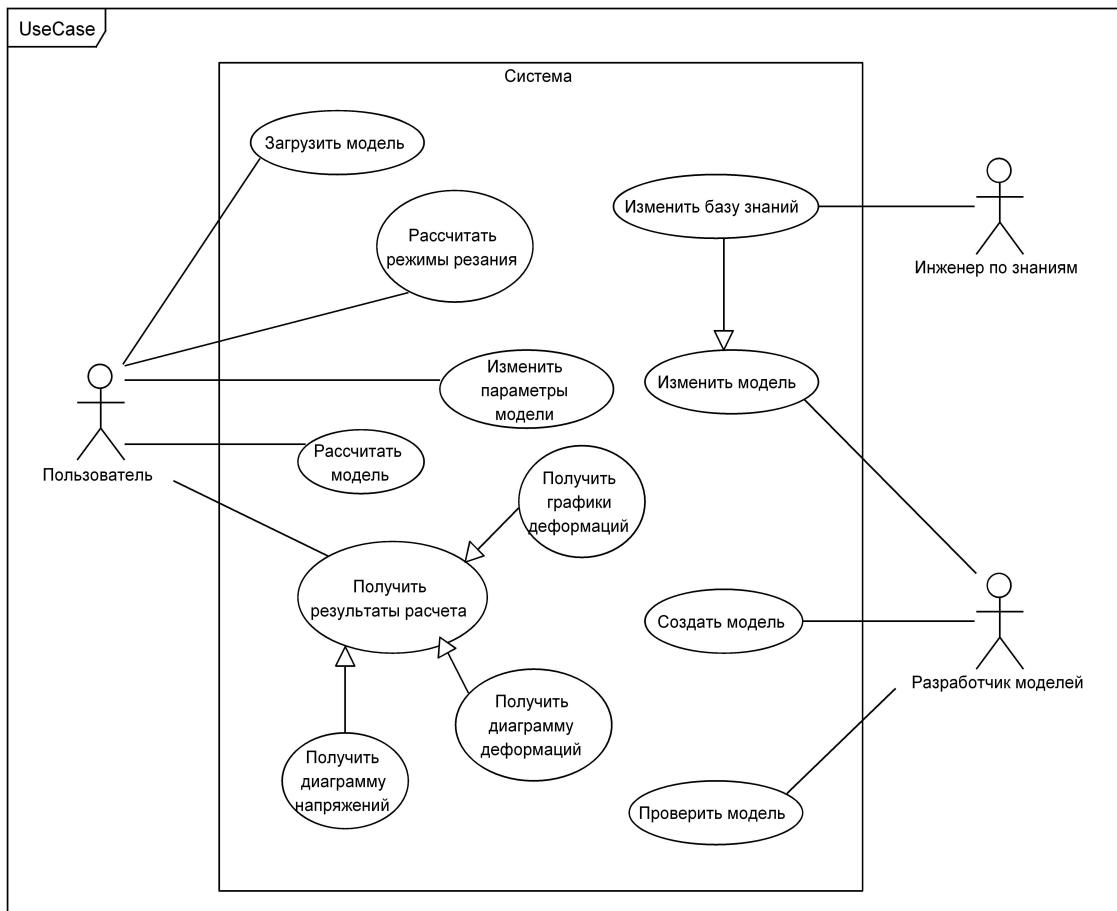


Рисунок 3 – Диаграмма вариантов использования

Загружаемая в систему модель должна быть предварительно подготовлена компетентным экспертом — разработчиком моделей. К обязанностям разработчика моделей относится также подготовка исходных данных в виде задания для выполнения САЕ системой. После подготовки модели эксперт проверяет её работоспособность и при необходимости устраняет дефекты, после чего предоставляет пользователю. Общая концепция системы предполагает, что все разработанные модели сохраняются в единую базу моделей, которую технолог использует для выбора расчетного случая, наиболее близкого заданному. Технолог не нуждается в самостоятельном программировании моделей, так как имеющийся состав информационной базы позволяет ему использовать готовые с учетом их высокой степени параметризации.

Составленная диаграмма вариантов использования изображена на ри-

сунке 3. Кроме описанных выше принципов на ней отображены зависимости обобщения между вариантами использования. Например, варианты использования “Получить диаграмму деформаций”, “Получить диаграмму напряжений” и “Получить графики деформаций” обобщены в “Получить результаты расчета”. Указанная диаграмма будет использована для выделения отдельных элементов системы в модули для последующей проработке.

5.2 Разработка структуры входных и выходных данных

5.2.1 Входные данные для анализа деформаций

На этапе эскизного проектирования необходимо определить какие входные данные будут использованы для расчета и в каком виде требуется представить результат работы системы. В контексте разрабатываемой системы моделью будет называться вся совокупность исходных данных, представленная в виде файла определенной структуры. Таким образом в данном разделе описан состав моделей, используемых системой.

Входные данные должны иметь минимум три основных компонента. На рисунке 4 показаны основные компоненты и один дополнительный – трехмерная OBJ модель, загружаемая для визуализации расчетного случая.

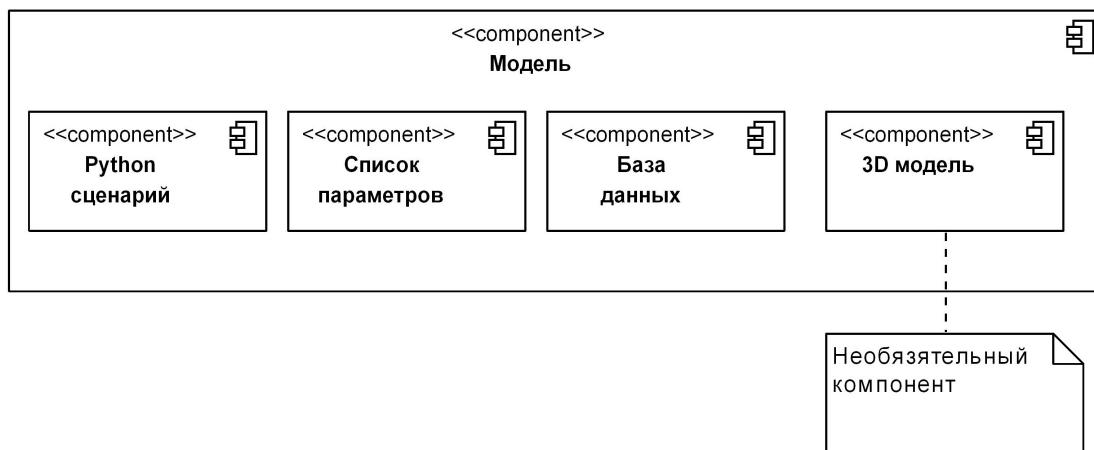


Рисунок 4 – Компоненты модели

Во-первых, сценарий построения расчетного случая в системе Abaqus. САЕ система Abaqus, как было указано ранее, использует язык Python для автоматизации расчетов. Выполнение определенной программы позволяет получить поля напряжений и узловых перемещений для запрограммированного сценария, а также записать их в виде графических или текстовых файлов для последующей обработки. Расчетный случай может быть параметризован любым количеством переменных, принимаемых в виде аргументов командной строки при вызове Python скрипта. Таким образом, один сценарий может быть применен многократно для деталей с общей формой, но различными значениями геометрических размеров.

Python относится к высокоуровневым объектно-ориентированным языкам программирования общего назначения. Он имеет динамическую типизацию, синтаксически простые конструкции ветвления, последовательного перебора элементов коллекций, обработки исключений, что делает его чрезвычайно легким в освоении и при этом гибким и мощным средством автоматизации.

Вторая составляющая модели это список параметров. Все параметры должны быть перечислены в отдельной структуре с указанием комментариев и размерности для того чтобы пользователь мог выбрать и изменить необходимые из них. Эта структура загружается системой вместе с остальными компонентами и отображается графическим интерфейсом. С точки зрения пользователя удобно иерархическое представление набора переменных, когда каждый параметр относится к какой либо подгруппе, например, “Кулачек”, которая в свою очередь входит в группу более высокого уровня. На самом верхнем уровне находится корневая группа “Модель”. Таким образом, список параметров будет логически представлен в виде дерева, внутренними узлами которого являются группы параметров, а листовыми вершинами сами параметры. На рисунке 5 показано абстрактное представление такого дерева произвольной конфигурации. Группы параметров обозначены латинскими буквами «G», а параметры – буквами «P».

Параметр с точки зрения пользователя представляется как кортеж трех

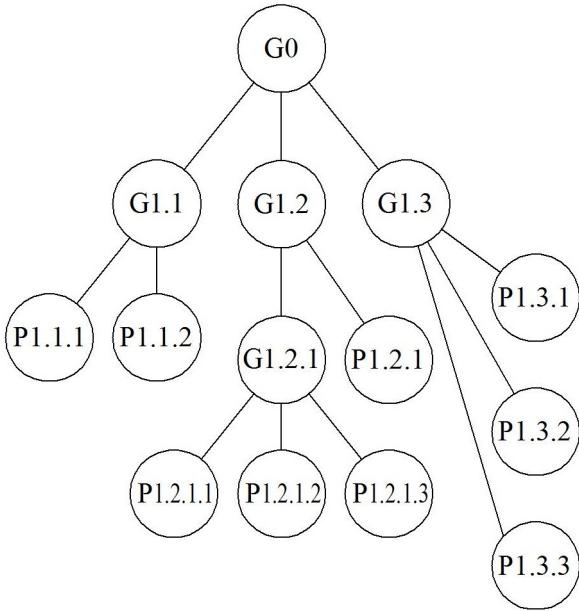


Рисунок 5 – Дерево параметров

элементов: $\langle N, C, V \rangle$, где N – название параметра (name), например, “Сила резания”, C – комментарий (commentary), поясняющий роль параметра и указывающий его размерность и V – значение (value) параметра, которое вводится пользователем. Группы параметров имеют только имя и служат контейнерами, обеспечивающие иерархическую вложенность.

Третья составляющая модели – база данных. База данных предназначена для предоставления справочных значений параметров, таких как свойства материалов. Например, пользователю известно, что деталь изготавливается из углеродистой стали определенной марки. Выбор заданного материала, занесенного в базу данных, инициирует автоматическое заполнение сопутствующих параметров: модуля упругости и коэффициента Пуассона.

Необязательным компонентом выступает трехмерная модель расчетного случая. Эта модель используется системой для визуализации схемы закрепления детали и её формы. Хотя трехмерное отображение может быть получено после загрузки модели в систему и выполнения программного сценария, тем не менее такой подход увеличит время загрузки и поэтому был признан неприемлемым. Дополнительно к недостаткам генерации

изображения после загрузки можно отнести зависимость от установленной САЕ системы. Если на ПК пользователя не установлена САЕ система, то пользователь не сможет не только произвести расчет (что ожидаемо), но и открыть модель для просмотра.

5.2.2 Входные данные для калькулятора режимов резания

Калькулятор режимов резания будет реализован отдельным модулем и для него входные данные вводятся отдельно. В соответствии с таблицами, приведенными в [3], было определено что для вычисления режимов резания по таблично-аналитической методике требуются входные данные, перечисленные в таблице 2. В указанной таблице также приведены возможные значения входных данных. Для модуля калькулятора режимов резания

Таблица 2 — Входные данные калькулятора режимов резания

Наименование	Значения
Диаметр обрабатываемой поверхности	Числовое в мм
Тип точения	Черновое, получистовое, чистовое
Вид обработки	Внешнее точение, растачивание, отрезание
Материал заготовки	Сталь, чугун
Вид заготовки	Поковка, отливка, прокат, полуфабрикат без корки
Материал инструмента	T15K10, T15L6, T30K4, BK8, BK6, BK4, быстрорежущая сталь
Период стойкости резца	20, 30, 40, 60, 90 мин
Шероховатость поверхности	По ГОСТ 2789-73
Радиус при вершине резца	Числовое в мм

не требуется разработки какой-либо особой структуры данных так как вся

информация непосредственно вводится пользователем с клавиатуры или выбирается из выпадающих списков.

5.3 Эскизное описание принципов работы

Работа системы поддержки принятия решения основывается на использовании вычислительных возможностей программы Abaqus. Рассмотрим подробнее последовательность взаимодействия пользователя, разрабатываемой системы и Abaqus.

На рисунке 6 приведена UML диаграмма последовательностей. На диаграмме показана последовательность обмена сообщениями для варианта использования “Изменить параметры модели” и “Рассчитать модель” а также вариант использования “Обновить модель”, который не показан на диаграмме прецедентов для упрощения восприятия.

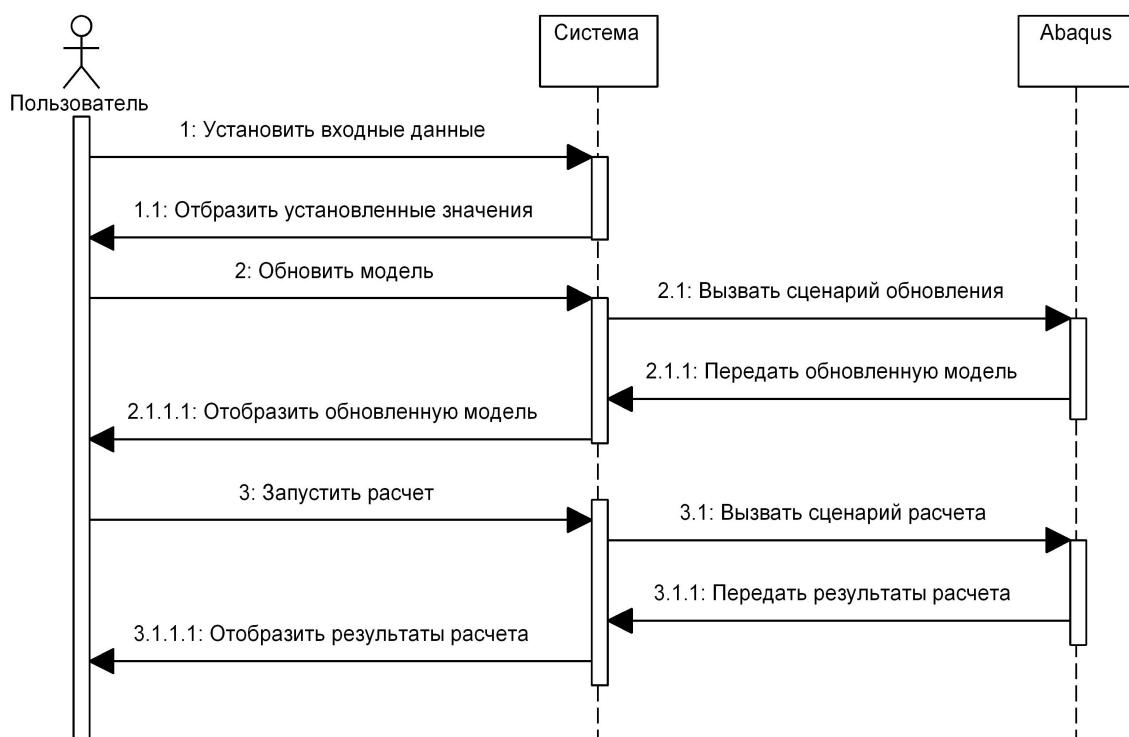


Рисунок 6 – Диаграмма последовательностей

Как видно из диаграммы, при задании пользователем значений па-

метров система сохраняет их в оперативной памяти и отображает в графическом интерфейсе. В случае если пользователю требуется обновить трехмерное отображение заготовки, программа обращается к Abaqus, вызывая специальный сценарий обновления, который представляет собой часть расчетного сценария, но вместо запуска вычислений сохраняет перестроенную геометрию в обменный файл, которым пользуется система для визуализации изменений. Аналогичный обмен сообщениями происходит при запуске расчета, но в этом случае уже вызывается полный сценарий, вычисляющий перемещения и напряжения в детали.

5.4 Модули

Разрабатываемая программа опирается на использование технологии NetBeans Platform, что позволяет при проектировании оперировать понятием модуля системы. В данном разделе описываются принципиальные подходы используемой технологии и их практическое применение.

5.4.1 Концепции NetBeans

В основе NetBeans Platform лежит идея, что каждая часть кода приложения принадлежит какому либо модулю [7]. Суть такого подхода заключается в том, что для каждого модуля определена собственная область видимости и коммуникация между модулями осуществляется только по определенным интерфейсам, которые описывают правила взаимодействия, но не раскрывают подробностей его реализации.

Концепция модуля в NetBeans платформе отличается от концепции Java пакета так как модуль имеет большую независимость от своей среды. Он объединяет несколько логически и функционально связных пакетов, инкапсулируя их. В некоторых случаях модуль может экспорттировать внутренний пакет для совместного использования.

Для передачи объекта от одного модуля к другому предусмотрена концепция поиска (“Lookup”) объектов. Модуль-поставщик публикует объект

определенного типа в экземпляр разделяемого класса (то есть класса, доступного обоим модулям), а модуль-приемник на основе механизма прослушивания или при непосредственной проверке в результате поиска объекта заданного типа получает ссылку на передаваемый объект и использует её для своего функционирования.

Таким образом, каждый модуль может быть разработан автономно, при условии соблюдения контрактов, описанных интерфейсами. Модуль должен быть слабо связан с самой системой, иметь лишь необходимый минимум зависимостей и не использовать специфических недокументированных особенностей кода, с которым происходит взаимодействие. В таком случае разработанная система отвечает требованиям простоты обслуживания и может быть легко расширена новыми функциональными возможностями.

5.4.2 Применение платформы

На эскизном этапе проектирования необходимо уточнить состав модулей и их предназначение. Вопросы реализации отдельных компонентов будут рассмотрены при техническом проектировании. В таблице 3 приведены разрабатываемые модули с указанием краткого описания по каждому из них. Название модуля отражает его предназначение. Первая часть названия “егемейкин.pete.” одинакова для всех разрабатываемых компонентов и составлена по подобию именования Java пакетов. Если название содержит в себе сокращение “api”, то такой модуль используется внешними модулями и предназначен для описания интерфейсов взаимодействия.

Таблица 3 – Состав модулей

Полное название	Предназначение
егемейкин.pete.api.core	Обмен объектами между модулями, обработка запуска системы и управление директориями рабочего пространства
егемейкин.pete.api.executor	Описание интерфейса вызова внешней САЕ системы

eremeykin.pete.api.model	Определение классов модели и параметра для совместного использования другими модулями
eremeykin.pete.actions	Делегирование соответствующим компонентам возможных действий пользователя, таких как “Обновить модель” и “Расчитать модель”
eremeykin.pete.calculator	Обеспечение функционала калькулятора режимов резания
eremeykin.pete.editor	Редактирование значений параметров модели
eremeykin.pete.launcher.abaqus	Конкретная реализация исполнителя Abaqus сценариев
eremeykin.pete.loader	Загрузка моделей в систему
eremeykin.pete.logger	Отладочные средства, логирование
eremeykin.pete.plotter	Загрузка результирующих данных и построение графиков деформаций
eremeykin.pete.reports	Генерация отчетных документов
eremeykin.pete.viewport	Трехмерное отображение модели детали

Для организации слабосвязанной системы модулей было выделено три компонента, которые экспортируют один или несколько пакетов, позволяя тем самым ввести общие для всей системы соглашения. Например, основополагающим понятием является модель. Класс модели должен быть использован во многих модулях, таких как модуль редактора параметров eremeykin.pete.editor или загрузчик eremeykin.pete.loader.

ГЛАВА 6

ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

6.1 Уточнение структуры входных данных

Ранее было определено, что исходные данные загружаются в систему в виде модели, содержащей всю необходимую информацию для работы системы. Определим подробнее формат хранения этой информации и способы её извлечения.

По результатам эскизного проектирования установлено, что большая часть информации должна быть сохранена в табличном виде. Между определенными фрагментами данных существуют логические связи, например, между названием материала заготовки и его физическими свойствами. Таким образом, было принято решение использовать базу данных (БД) как источник информации о загружаемой модели.

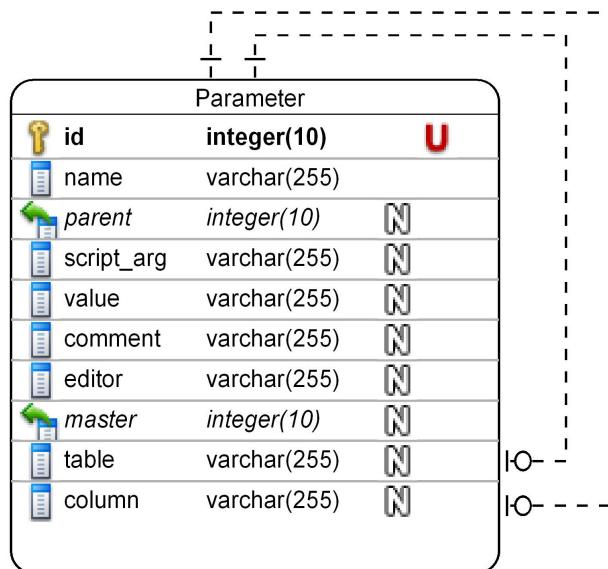


Рисунок 7 – ER диаграмма

В качестве альтернативного варианта реализовано хранение моделей в табличном файле Excel. Этот способ существенно проще с точки зрения пользователя и разработчика моделей так как данные могут быть отредактированы в графическом интерфейсе Microsoft Office или аналогичных

программ. Табличная структура файлов Excel в целом похожа на способ представления данных в БД, поэтому будет разработан один способ записи модели для двух вариантов физической реализации.

На рисунке 7 изображена ER диаграмма, показывающая состав атрибутов параметров модели и их отношения. Параметр имеет 10 атрибутов в общем случае, некоторые из которых могут иметь значение NULL (помечены буквой N). Группы параметров, о которых было сказано на этапе эскизного проектирования, также описываются сущностью **Parameter**, но имеют определенные особенности.

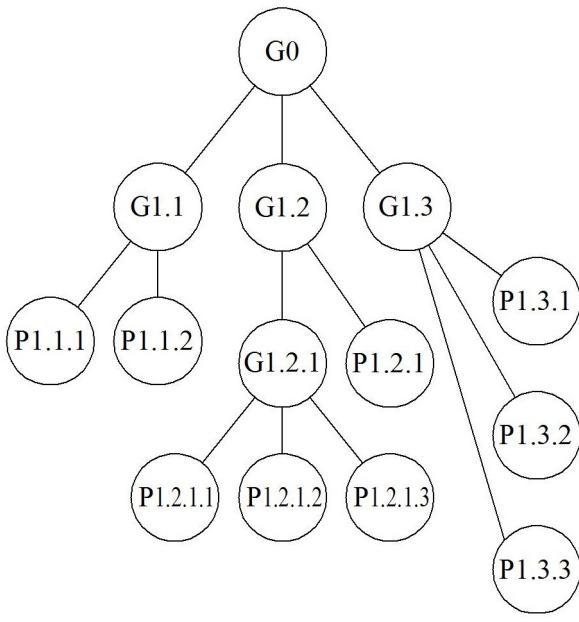
Атрибут **id** выступает в роли первичного ключа и имеет целочисленное уникальное значение (помечено буквой U). Он не отображается в пользовательском интерфейсе программы и необходим для построения древовидной структуры с помощью ссылок.

За отображение названия параметра в графическом интерфейсе отвечает атрибут **name**, имеющий строковое значение. Название кратко характеризует роль параметра в модели. Примерами имени являются “Модуль упругости”, “Ширина” для параметров, а для групп – “Кулакки”, “Деталь”. Для данного атрибута не допускается указывать NULL значение, так как параметр или группа обязательно должны иметь название.

Особую роль в построении иерархии параметров играет атрибут **parent**. Этот атрибут указывает на своего родителя более высокого уровня. Родителем параметра может выступать только группа параметров, как показано на рисунке 5. Например, на этом рисунке параметр P1.1.1 входит в состав группы G1.1. Тогда значение **parent** для этого параметра будет содержать **id** группы G1.1. Аналогично для групп нижних уровней в роли родителя выступает группа верхнего уровня. Корневая группа имеет значение **parent=NULL**, что допустимо согласно приведенной ER диаграмме. Для приведенного на рисунке 5 примера составлена таблица 4.

Для передачи действительных значений параметров в Python скрипт используется атрибут **script_arg**, который указывает принадлежность конкретного значения к определенному параметру. Пример команды, вызы-

Таблица 4



id	name	parent
0	G0	NULL
1	G1.1	0
2	G1.2	0
3	G1.3	0
4	P1.1.1	1
5	P1.1.2	1
6	G1.2.1	2
7	P1.2.1	2
8	P1.3.1	3
9	P1.3.2	3
10	P1.3.3	3
...

вающей запуск скрипта можно записать следующим образом:

```
abaqus cae noGUI=script.py inner=0.028 outer=0.03 ...
```

Из примера видно, что для ассоциации значения параметра используются ключевые слова, такие как `inner` и `outer` с помощью которых скрипт распознает за что отвечает следующее за символом “=” значение. Эти ключевые слова записываются в колонке `script_arg`. Значение `NULL` служит для обозначения групп параметров или каких-либо отдельных вспомогательных параметров, значение которых не требуется для работы Python скрипта.

Атрибут `value` служит для выставления значения по-умолчанию и может быть впоследствии использован при сохранении состояния модели в файл. Этот атрибут выводится на экран после загрузки модели и изменяется пользователем в графическом интерфейсе. Для групп и параметров, не требующих значения по-умолчанию, `value` устанавливается равным `NULL`.

Для вспомогательных пояснений относительно роли конкретного па-

раметра предусмотрен атрибут `comment`. Как следует из названия в нем хранятся строковые комментарии. Хорошей практикой считается указание размерности в этой графе помимо дополнительной информации. Есть возможность не использовать комментарии, если в них нет необходимости. При этом атрибут `comment` принимает значение `NULL`.

Колонка `editor` описывает тип редактируемого параметра. Если параметр не может быть изменен пользователем, то в поле `editor` записывается `NULL`, это же значение присваивается группам параметров. Для редактируемых параметров предусмотрено 3 допустимых значения: `text`, `auto` и `cbox`. Значение `text` соответствует простому текстовому полю для ввода числового или строкового значения. Например, такое поле используется для описания параметра “Внешний радиус”. Если указано значение `auto`, то в этом случае величина параметра должна быть определена автоматически в зависимости от другого параметра. Примером автоматически заполняемого параметра является “Модуль упругости”. Для обозначения параметра, редактируемого выпадающим списком с предлагаемым набором значений служит `cbox`.

Принимая во внимание зависимость некоторых параметров от других (например, модуль упругости зависит от материала), было принято решение отмечать такую зависимость в таблице параметров при помощи атрибута `master`. Этот атрибут указывает на `id` главного параметра, который управляет текущим. То есть если в поле `master` параметра “Модуль упругости” будет указан `id` параметра “Название материала”, то модуль упругости будет выставляться в зависимости от выбранного материала. Но для этого необходимо сопоставить название материала и его параметры. Эту роль выполняет дополнительная таблица `materials`. Для указания таблицы соответствия (в роли неё могут выступать и другие таблицы, не обязательно `materials`) используется атрибут `table`, а атрибут `column` указывает на колонку по которой выбирается значение.

6.2 Чтение входных данных

Входные данные могут быть извлечены из SQL базы данных или из Excel таблиц. При разработке способа их чтения следует предусмотреть и другие варианты хранения на случай расширения системы, поэтому чтение входных данных рассмотрено подробно.

6.2.1 Подключение базы данных

Для извлечения информации из баз данных при помощи языка Java существует стандартный способ, называемый Java Data Base Connectivity (JDBC). Последняя версия JDBC 4.2 описывается в документе Java Specification Request (JSR) 221.

Преимущество этого стандарта заключается в том, что при условии рационального программирования SQL запросов, он позволяет создать программу, не зависящую от конкретной реализации системы управления базами данных (СУБД). Стандарт JDBC основывается на использовании так называемых JDBC драйверов. Каждый производитель, поддерживающий работу со своими СУБД поставляет как минимум один драйвер, который, согласно стандарту, содержит определенную реализацию интерфейса `java.sql.Driver`. Таким образом достигается отсутствие непосредственной связи кода программы и кода JDBC драйверов различных производителей СУБД. Например, если изначально программа была ориентирована на использование СУБД Oracle Database и при этом не были задействованы особенности диалекта Oracle SQL, то в процессе последующего развития к программе могут быть подключены и другие системы управления базами данных (например, MySQL), что потребует добавления соответствующего JDBC драйвера.

6.2.2 Выбор СУБД

Определим СУБД которая будет использована на начальных стадиях развития интегрированной системы поддержки принятия решений. Так как

на момент разработки ещё не существует соответствующей информационной среды, то будет использована встраиваемая СУБД, работающая на том же компьютере что и сама система. При последующем развитии системы встраиваемая СУБД может быть заменена на клиент-серверную, так как механизм JDBC взаимодействия остается общим.

Простота, удобство и функциональность определили выбор SQLite. СУБД относится к встраиваемым так как не требует запуска отдельного серверного процесса, а подключается в виде библиотеки. Все данные сохраняются в отдельный файл, который удобно передавать для использования на других компьютерах без необходимости настраивать подключение к серверу, как в случае клиент-серверных СУБД. Тем не менее сохраняются все преимущества реляционных баз данных.

6.2.3 Реализация шаблона DAO

В программировании существуют общепринятые методики решения определенных типичных проблем. Эти решения описываются в виде шаблонов проектирования. Благодаря применению шаблонных решений можно существенно снизить трудоемкость создания программных продуктов и повысить эффективность их обслуживания.

Для решения проблемы чтения данных рекомендуется [9] использовать шаблон Data Access Object (DAO). Этот шаблон подразумевает введение специального объекта доступа к данным для абстракции и инкапсуляции механизмов взаимодействия с источником информации. Благодаря унификации интерфейса доступа к различным хранилищами достигается большая гибкость архитектуры приложения. В случае, если потребуется реализовать извлечение данных из некоторого нового источника, не предусмотренного изначально техническим заданием, такой функционал можно обеспечить добавлением одного соответствующего класса, при этом не придется изменять уже написанный код.

Реализация шаблона DAO включает четыре компонента. Первый компонент – объект, которому требуется обеспечить доступ к некоторым данным.

Такой объект называют клиентом или бизнес объектом. Второй компонент представляет собой собственно объект доступа к данным (DAO). Он играет роль универсального адаптера, задача которого состоит в скрытии структуры любого хранилища данных, к которому требуется подключиться и предоставлении универсального интерфейса доступа клиента к содержащейся в хранилище информации. Носителем данных выступает третий компонент, представленный в виде класса, экземпляр которого возвращается клиенту. Последним компонентом является сам источник данных.

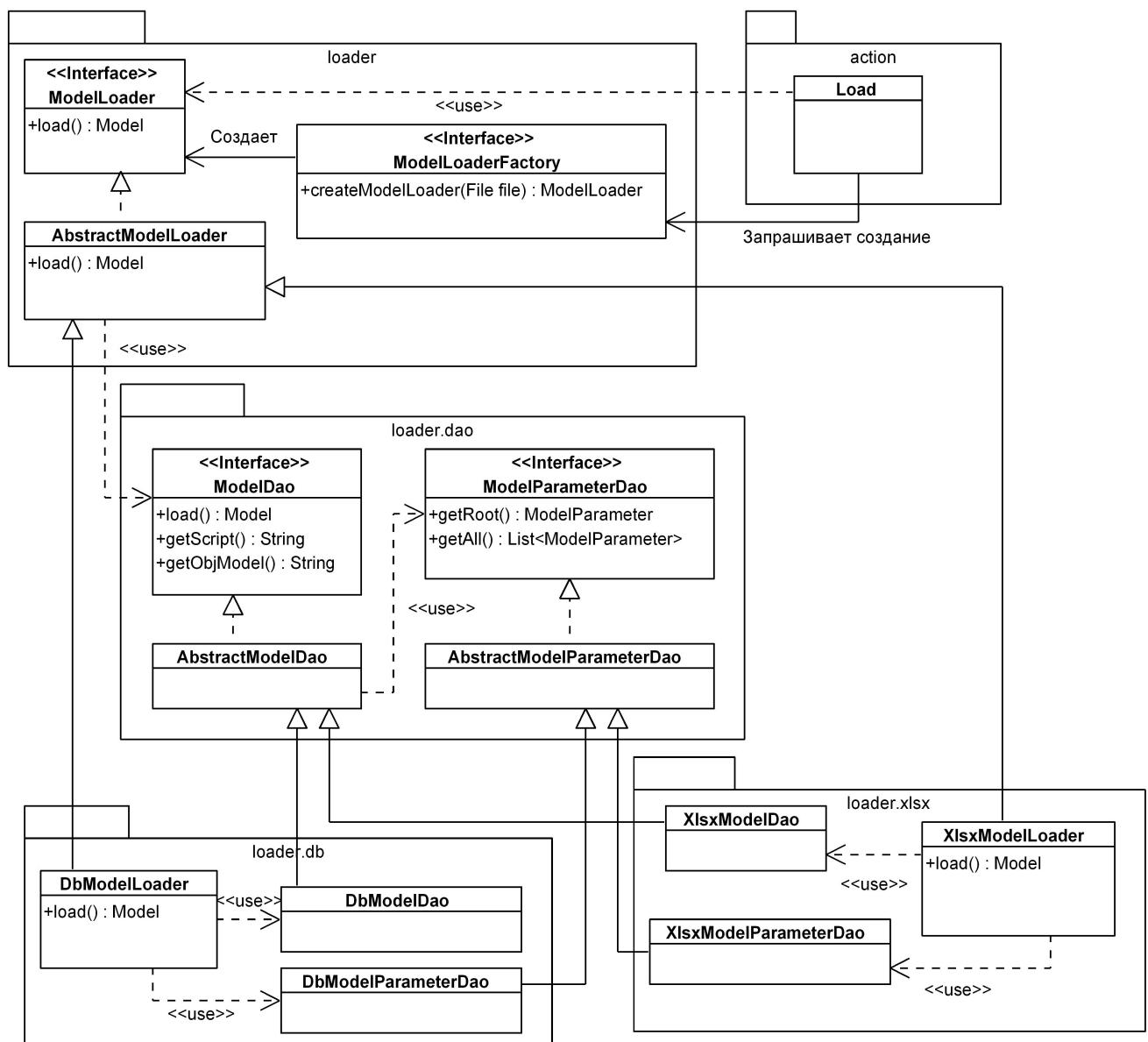


Рисунок 8 – Диаграмма классов

В случае интегрированной системы поддержки принятия решений необходимо осуществить чтение информации из SQL базы данных и из Excel файла. Для этого была разработана структура классов, показанная на рисунке 8.

В качестве клиента выступает класс, описывающий действия при нажатии на кнопку “Загрузить модель”. Этот класс назван Load, он реализует интерфейс `java.awt.event.ActionListener`, объявленный в JDK. Вызовом метода `createModelLoader(File file)` класс Load запрашивает создание экземпляра `ModelLoader`. В зависимости от типа файла (.sqlite или .xlsx) клиенту предоставляется та или иная реализация `AbstractModelLoader` через ссылку типа `ModelLoader`. Конкретный класс возвращаемого объекта не играет роли для клиента, так как известно, что любая реализация интерфейса `ModelLoader` имеет метод `load()`, возвращающий объект модели `Model`. Именно этот объект и требуется для работы клиента и отображения загруженной модели.

Имеются две реализации абстрактного класса `AbstractModelLoader` в соответствии с требуемыми типами хранилищ – это `DbModelLoader` и `XlsxModelLoader`. Первая реализация используется для чтения информации из базы данных и использует JDBC драйвер для выполнения SQL запросов. Вторая реализация считывает данные из xlsx файла при помощи библиотеки с открытым исходным кодом Apache POI.

Собственно объект доступа к данным имеет тип `ModelDao` для чтения всей модели и `ModelParameterDao` для чтения отдельного параметра из модели. `ModelDao` и `ModelParameterDao` являются интерфейсами, а общая функциональность для всех видов хранилищ вынесена в абстрактные классы `AbstractModelDao` и `AbstractModelParameterDao`.

Благодаря применению шаблона DAO достигается возможность наращивания функционала программы. Например, в данную структуру классов легко добавить возможность загрузки модели из xml файла. Для этого потребуется написать класс `XmlModelLoader` и соответствующие DAO классы, которые расширяют имеющиеся абстрактные классы.

6.3 Построение МКЭ модели

Для демонстрации работоспособности системы будет построена модель обработки тонкостенного цилиндра. Эскиз детали изображен на рисунке 9, а операционный эскиз на рисунке 10 .

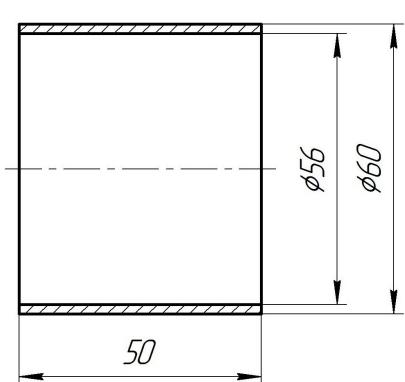


Рисунок 9 – Эскиз детали

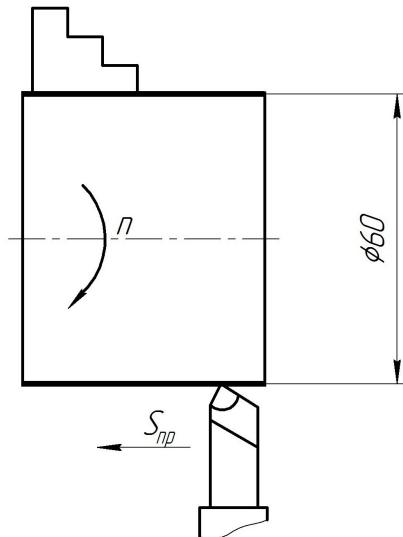


Рисунок 10 – Операционный эскиз

6.3.1 Выбор типа конечных элементов

Так как заданная деталь относится к типу тонкостенных, то наиболее подходящим элементом разбиения является оболочковый. Как рекомендует документация Abaqus для общего применения при достаточно больших напряжениях следует назначить элемент типа S4R. Название этого элемента расшифровывается как **S**hell, **4**-node, **R**educed integration, то есть оболочковый, 4-х узловой, с уменьшенным числом точек интегрирования .

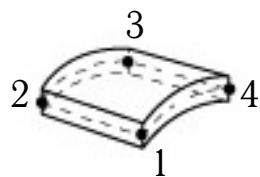


Рисунок 11 – Оболочковый элемент S4R

Для нахождения матрицы жесткости элементов требуется вычислять интегралы вида $\int_{-1}^1 f(\xi) d\xi$ или для двухмерного случая $\int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta$. Аналитически вычислить значения указанных выражений в общем случае не удается, поэтому применяют численные методы. Интеграл заменяется суммой:

$$\int_{-1}^1 f(\xi) d\xi \approx \sum_{i=1}^n a_i f(\xi_i) \quad (6.1)$$

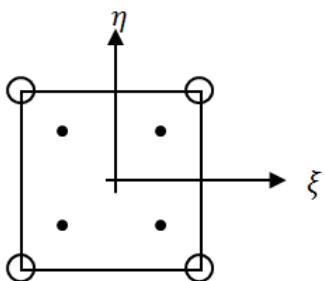
где ξ_i – точки интегрирования;

a_i – весовые коэффициенты.

Точки интегрирования выбираются исходя из конкретного метода интегрирования, но всегда при $n \rightarrow \infty$ должно выполняться условие:

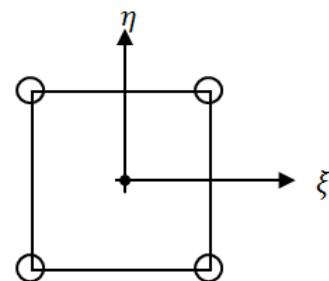
$$\sum_{i=1}^n a_i f(\xi_i) \rightarrow \int_{-1}^1 f(\xi) d\xi$$

В МКЭ желательно применять такие методы выбора точек интегрирования и весовых коэффициентов, которые обеспечивали бы наибольшую точность при наименьшем числе точек интегрирования ввиду трудоемкости вычисления значения $f(\xi_i)$. Поэтому применяется метод Гаусса, согласно которому точки ξ_i и коэффициенты a_i выбираются таким образом, чтобы формула (6.1) представляла собой точное равенство для $f(\xi) = 1, \xi, \xi^2, \dots, \xi^p$. Так как число варьируемых величин в методе Гаусса равно $2n$, где n – число точек интегрирования, то максимальный порядок интегрирования $p \leq 2n - 1$.



(а) Полное интегрирование S4

○ Узлы
• Точки интегрирования



(б) Сокращенное интегрирование S4R

Рисунок 12 – Точки интегрирования

В случае плоского четырехузлового элемента подынтегральная функция $f(\xi, \eta)$ является квадратичной по координатам ξ и η , т.е. $p = 2$ и $n = 2$.

Координаты две, поэтому говорят, что интегрирование осуществляется по 2×2 точкам. Такое интегрирования называется полным (рисунок 12а). Для уменьшения объема вычислений используется сокращенное число точек интегрирования. Для элемента S4R используется одна точка с координатами $\xi = 0, \eta = 0$ как показано на рисунке 12б.

Кулачки моделируются пространственным элементом C3D8R – Continuum stress/displacement, **3D** – three-dimensional, **8-node**, **Reduced integration**, то есть 8-ми узловой элемент для моделирования непрерывных полей напряжений/перемещений с уменьшенным числом точек интегрирования.

6.3.2 Допущения

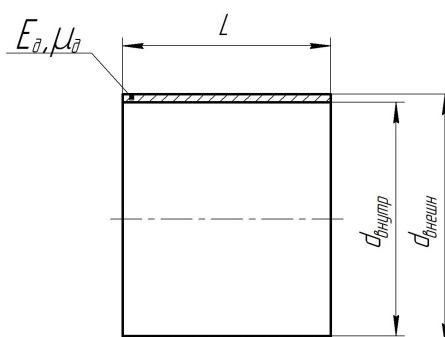
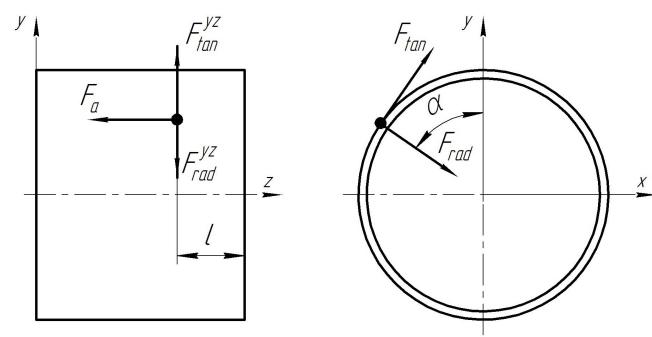
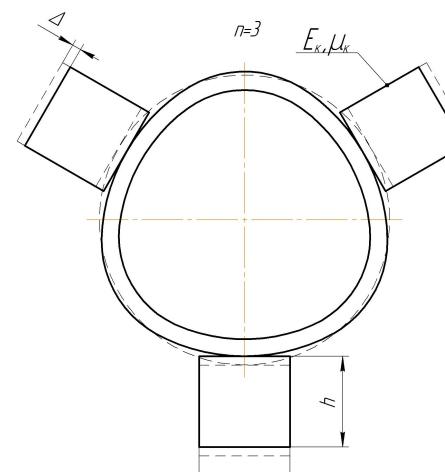
При построении модели воздействие резца на деталь учитывается в виде точечной трехкомпонентной силы, нагрев детали в рассмотрение не принимается. Считается, что сила резания в процессе обработки постоянна, глубина резания не изменяется, материал заготовки однородный. Кулачки моделируются в виде жестких ($E_{кулачка} \gg E_{детали}$) прямоугольных параллелепипедов, равномерно со всех сторон обхватывающих заготовку.

6.3.3 Параметры модели

Для повышения эффективности работы разработчика моделей следует заранее составлять хорошо параметризованные модели. При достаточно большом числе параметров появляется возможность использовать одну модель для расчета нескольких деталей.

В таблице 5 приведены все изменяемые параметры для моделирования процесса обработки заданной детали, сгруппированные в трех категориях. Для детали параметрами полностью задается геометрические размеры и свойства материала. Аналогичные параметры выделены и для кулачков, но в эту категорию также добавлена величина радиального смещения и количество кулачков. Сила резания задается координатами точки приложения в цилиндрической системе координат и тремя компонентами.

Таблица 5 — Параметры модели

Обозн.	Название	Схема
Деталь		
L	Длина детали	
$d_{\text{внутр}}$	Внутренний диаметр	
$d_{\text{внешн}}$	Внешний диаметр	
E_d	Модуль упругости материала детали	
μ_d	Коэффициент Пуассона материала детали	
Сила резания		
F_a	Осевая компонента	
F_{tan}	Тангенциальная компо- нента	
F_{rad}	Радиальная компонента	
l	Осьное положение точки приложения	
α	Угловое положение точки приложения	
Кулачки		
Δ	Радиальное смещение	
n	Количество кулачков	
w	Ширина кулачка	
h	Высота кулачка	
t	Толщина кулачка	
E_k	Модуль упругости материала кулачков	
μ_k	Коэффициент Пуассона материала кулачков	

6.3.4 Модель в Abaqus

На рисунке 13 показана построенная модель и её разбиение на конечные элементы. Построение модели полностью автоматизировано благодаря применению встроенного в Abaqus языка сценариев.

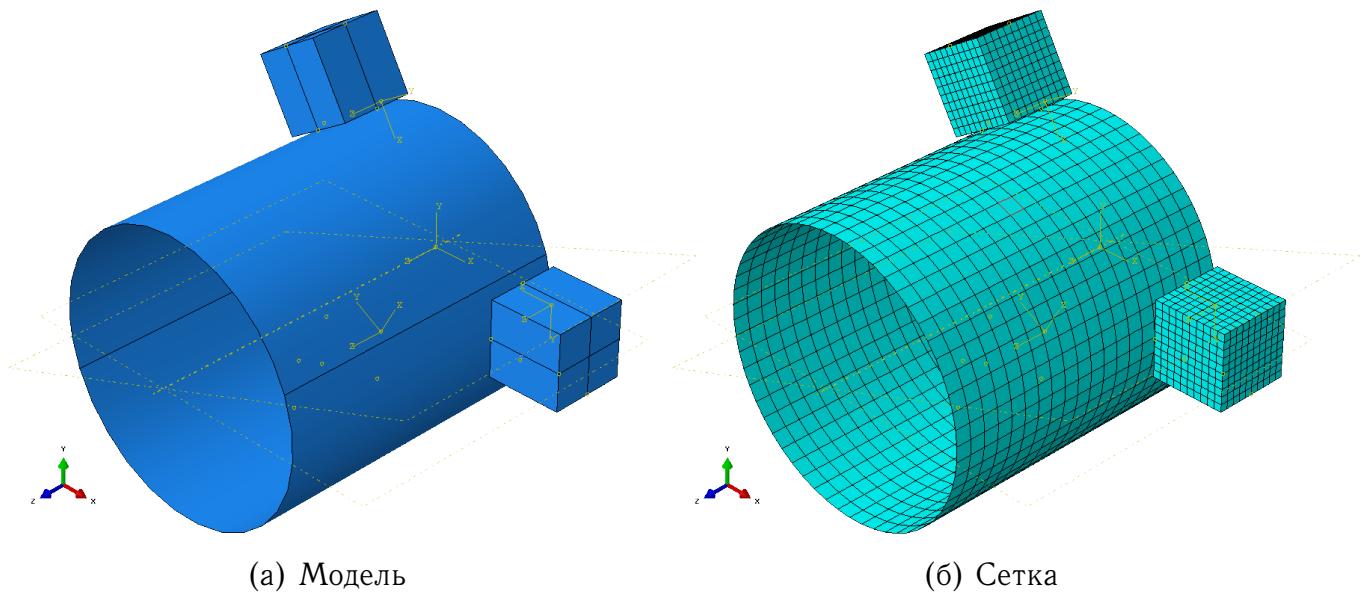


Рисунок 13 – Модель в Abaqus

Для приложения силы резания в точке, соответствующей заданным координатам, применяется вспомогательное сечение плоскостью, расположенной под углом α к центральной плоскости кулачка.

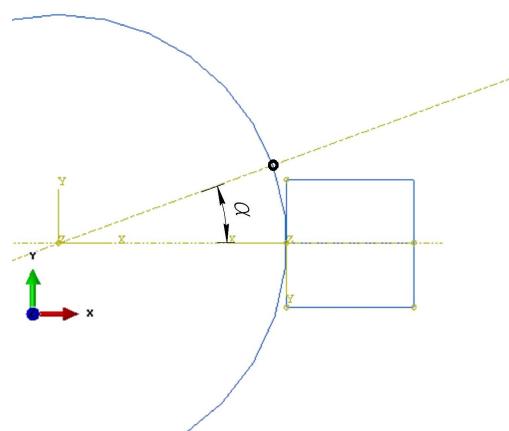


Рисунок 14 – Вспомогательное сечение

На рисунке 15 показаны поля напряжения и деформаций. На диаграмме напряжений отчетливо видна точка приложения силы резания. Как было сказано выше, жесткость кулачков намного больше жесткости самой детали, поэтому изменения напряжений и деформаций на них не могут быть отображены в данном масштабе.

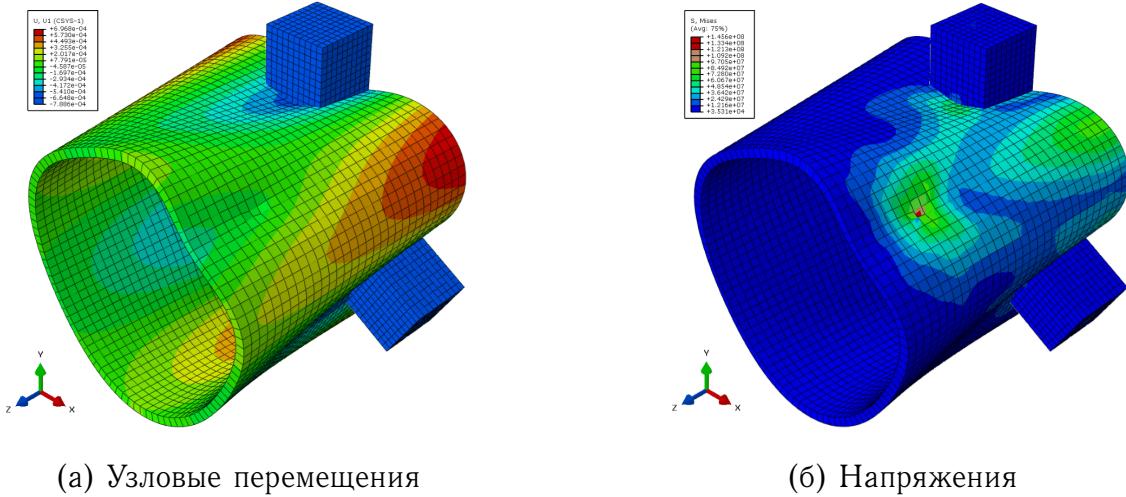


Рисунок 15 – Результаты расчета

6.4 Разработка калькулятора резания

Калькулятор режимов резания разрабатывается в виде отдельной подсистемы, физически реализованной как модуль платформы NetBeans. Он служит для определения начального приближения значений сил резания и предполагает использование таблично-аналитической методики, описанной в [3]. Структура исходных данных, определенная на этапе эскизного проектирования не нуждается в уточнении.

Рассматриваемая таблично-аналитическая методика использует большое количество эмпирических коэффициентов, которые могут на самом деле отличаться от приведенных справочных значений. Поэтому было принято решение организовать вычисление режимов резания по гибкой технологии, в основе которой лежит использование баз знаний.

6.4.1 Библиотека Drools

Для Java существует бесплатная библиотека с открытым исходным кодом Drools. Эта библиотека позволяет осуществлять логический вывод, основанный на использовании продукционных правил. На рисунке 16 показан алгоритм работы машины вывода Dools согласно официальной документации [10]. Рабочая память сопоставляется с базой правил для определения конфликтного множества, то есть набора правил, которые могут быть выполнены в данный момент. При помощи заданной стратегии разрешения конфликтов из конфликтного множества определяется единственное правило, которое затем исполняется. Процесс повторяется пока существует хотя бы одно правило для выполнения или пока не произойдет выход при выполнении одного из правил.

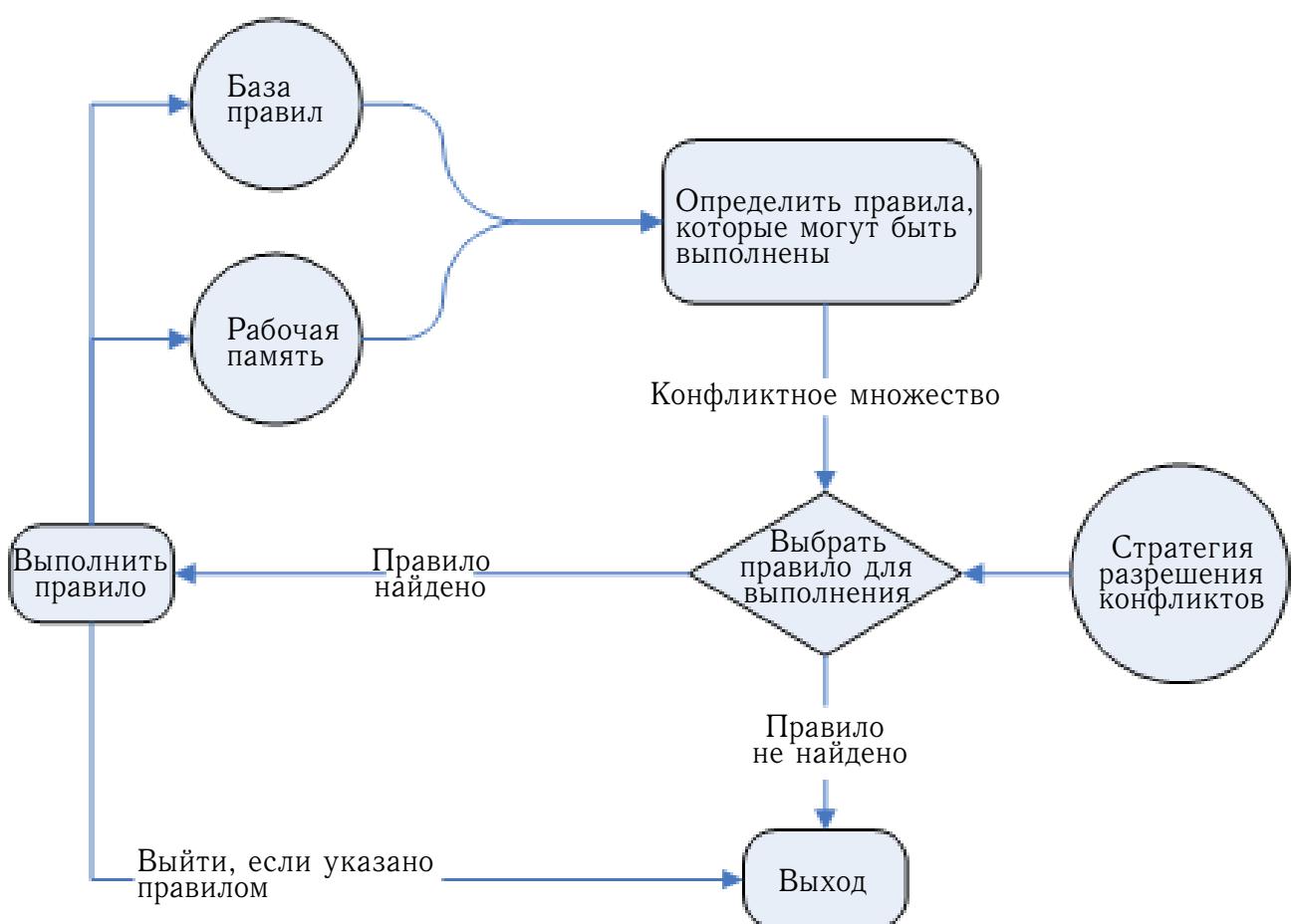


Рисунок 16 – Алгоритм работы Drools

6.4.2 Запись продукционных правил

Продукционные правила могут быть записаны в двух видах. Первый способ заключается в последовательном перечислении в текстовом файле всех правил, составленных на специализированном языке DRL - Drools Rule Language. Этот язык позволяет записывать правила в следующем виде:

```
1 rule "name"
      attributes
3   when
4     LHS
5   then
6     RHS
7 end
```

В приведенном фрагменте используются обозначения: `name` - имя правила; `LHS` - левая часть правила, то есть предусловие, которое должно быть соблюдено, чтобы правило сработало; `RHS` - правая часть правила, содержащая код, который выполняется в случае срабатывания правила; `attributes` - специальные атрибуты, модифицирующие работу правил. Например, формула $N = P_z V / 1020 \cdot 60$ для вычисления мощности резания, в виде продукционного правила представляется следующим образом:

```
1 rule "Calculate N"
2   salience -140
3   lock-on-active true
4   when
5     step : TurningStepBean ()
6   then
7     modify(step){setN(step.P*step.v/(1020*60))}
8 end
```

Приведенное правило имеет низкий приоритет (задается после ключевого слова `salience`), поэтому оно будет проверяться только после того как будут выбраны все необходимые эмпирические коэффициенты. Приоритет правил позволяет упорядочить их таким образом, чтобы не возникало недостатка данных. Это можно пояснить на следующем примере: формула вычисления мощности резания $N = P_z V / 1020 \cdot 60$ требует известных значений силы P_z и скорости резания V , поэтому соответствующее продукционное правило имеет приоритет ниже, чем приоритет формул $V = \frac{C^v}{T^m t^x s^y} K_{MV} K_{MP} K_{IV}$ и $P_z = 10 C_p t^x s^y V^n K_{MP}$.

RuleTable Depth of cut					
CONDITION	CONDITION	CONDITION	LOCK-ON-ACTIVE	PRIORITY	ACTION
step : TurningStepBean()					
kind in \$param	type == \$param	diameter > \$1 , diameter <= \$2			modify(step) {setT(\$param)}
Вид обработки	Тип точения	Диаметр			Глубина резания
(EXTERNAL_TURNING, GROOVING, PARTING)	ROUGH	6, 10	true	-1	0.85
		10, 18	true	-2	0.85
		18, 30	true	-3	1.05
		30, 50	true	-4	1.05
		50, 80	true	-5	1.25
		80, 100	true	-6	1.35
	SEMI_ROUGH	6, 10	true	-7	0.65
		10, 18	true	-8	0.7
		18, 30	true	-9	0.8
		30, 50	true	-10	0.85
		50, 80	true	-11	1.1
		80, 100	true	-12	1.15
	FINISH	0, 10	true	-13	0.45
		10, 18	true	-14	0.55
		18, 30	true	-15	0.65
		30, 50	true	-16	0.75
		50, 80	true	-17	0.85
		80, 100	true	-18	0.9
		10, 18	true	-19	0.4

Рисунок 17 – Продукционные правила в виде Excel таблиц (фрагмент)

Расчет режимов резания по формулам основывается на значениях предварительно выбранных эмпирических коэффициентов. Для назначения коэффициентов используется второй вид записи знаний в Drools - в виде Excel таблиц. Фрагмент таблицы для определения глубины резания показан на рисунке 17. Столбцы таблицы решений Drools могут быть отмечены ключевыми словами, в данном случае используются следующие:

CONDITION – условие, то есть такой столбец соответствует левой части продукционного правила

LOCK-ON-ACTIVE – запрещает выполнение других правил во время выполнения данного правила

PRIORITY – приоритет, определяет порядок проверки правил

ACTION – действие, соответствует правой части правила

В приведенной таблице запись `step : TurningStepBean()` означает, что для выполнения постусловия необходимо, чтобы в рабочей памяти находился объект класса `TurningStepBean`. Если такой объект будет найден, то он будет доступен по ссылке `step`. Ячейки следующей строки проверяют свойства найденного объекта для столбцов условия или определяют действие над объектом в случае столбца действия. Таким образом, запись `kind in $param` обозначает, что найденный объект должен удовлетворять определенному множеству значений по свойству `kind`. Соответствующее множество допустимых значений указывается в круглых скобках в ячейках ниже. Второй слева столбец предназначен для проверки типа точения, его условие записывается как `type==$param`. Третий столбец описывает ограничения на диаметр заготовки. Например, для первой строки диаметр заготовки должен лежать в диапазоне от 6 до 10 мм. Столбцы **LOCK-ON-ACTIVE** и **PRIORITY** управляют выполнением правил. Последний столбец описывает действие над фактами рабочей области. В данном случае для объекта `step` устанавливается глубина резания при помощи инструкции `modify(step){setT($param)}`.

Фактически все строки Excel таблицы на рисунке 17 во время выполнения компилируются в отдельные правила на языке DRL. Первая строка указанной таблицы компилируется в правило на языке DRL следующим образом:

```
1 rule "Depth of cut_10"
2     salience -1
3     lock-on-active true
4     when
5         step : TurningStepBean(kind in (EXTERNAL_TURNING,
6                 GROOVING, PARTING), type==ROUGH,
7                 diameter>6 , diameter<=10)
8     then
9         modify(step) {setT(0.85)}
10 end
```

ГЛАВА 7

РАСЧЕТ ЗАТРАТ НА РАЗРАБОТКУ ИНТЕГРИРОВАННОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ

7.1 Введение

В данной главе сформулированы этапы разработки, произведен расчет трудоемкости, установлен состав исполнителей, определена продолжительность выполнения отдельных этапов и вычислены затраты на создание системы поддержки принятия решения.

7.2 Организация и планирование процесса разработки

Разработка интегрированной системы поддержки принятия решения является сложным процессом, стадии которого определяются ГОСТом Единой Системы Программной Документации [11]. Согласно указанному документу всего выделяется 5 стадий, их подробное описание приведено в таблице 6.

Основными факторами, определяющими трудоемкость создания интегрированной системы являются: степень новизны разрабатываемого программного комплекса, сложность алгоритма его функционирования, объем используемой информации, вид её представления и способ обработки, уровень языка программирования [12]. Определим принадлежность разрабатываемого продукта к группам по каждому из указанных факторов.

По степени новизны интегрированная система может быть отнесена к группе «Б»: она не имеет аналогов, но и не требует разработки принципиально новых методов создания.

По степени сложности систему можно отнести к группе продукции, реализующей алгоритмы стандартных методов решения задач, то есть к группе 3. Непосредственная реализация метода конечных элементов не входит в рамки разработки продукта, а включение модуля оптимизации в состав системы на текущем этапе не предполагается.

Таблица 6 — Стадии разработки

Стадия разработки	Состав выполняемых работ
Техническое задание	Постановка задач, выбор критериев эффективности. Разработка технико-экономического обоснования разработки. Предварительный выбор методов выполнения работы. Разработка календарного плана выполнения работ
Эскизный проект	Предварительная разработка структуры входных и выходных данных. Разработка общего описания алгоритмов реализации решения задач. Разработка пояснительной записки. Согласование и утверждение эскизного проекта
Технический проект	Разработка алгоритмов решения задач. Разработка пояснительной записки. Согласование и утверждение технического проекта. Уточнение структуры, анализ и определение формы представления входных и выходных данных. Выбор конфигурации технических средств.
Рабочий проект	Комплексная отладка и сдача в опытную эксплуатацию. Разработка проектной документации. Разработка, согласование программы и методики испытаний. Предварительное проведение всех видов испытаний.
Внедрение	Подготовка и передача программной документации для сопровождения с оформлением соответствующего акта. Передача программной продукции в фонд алгоритмов и программ. Проверка алгоритмов и программ решения задач, корректировка документации после опытной эксплуатации программного продукта.

Как показал анализ литературы [3], исходная информация представлена как в виде формул, так и в виде разнородных таблиц. При этом требуется учитывать взаимное влияние данных, содержащихся в этих таблицах. Таким образом, разрабатываемую программу по виду представления исходной информации относят к группе 11. По структуре выходных документов система принадлежит группе 22 - требуется вывод на печать одинаковых документов, представляющих собой отчеты о моделировании.

7.3 Расчет трудоемкости этапов

7.3.1 Расчет трудоемкости разработки технического задания

Трудоемкость разработки технического задания может быть рассчитана по формуле:

$$\tau_{\text{TZ}} = T_{\text{PZ}}^3 + T_{\text{PP}}^3, \quad (7.1)$$

где τ_{TZ} – трудоемкость разработки технического задания на создание программного продукта, чел.-дни;

T_{PZ}^3 – затраты времени разработчика постановки задачи на разработку ТЗ, чел.-дни;

T_{PP}^3 – затраты времени разработчика программного обеспечения на разработку ТЗ, чел.-дни.

Значения величин T_{PZ}^3 и T_{PP}^3 рассчитывают по формулам:

$$T_{\text{PZ}}^3 = t_3 \cdot K_{\text{PZ}}^3; \quad T_{\text{PP}}^3 = t_3 \cdot K_{\text{PP}}^3,$$

где t_3 – норма времени на разработку ТЗ на программный продукт, чел.-дни; $t_3 = 69$ чел.-дни, согласно таблице 2, приведенной в [12].

K_{PZ}^3 – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задачи на стадии ТЗ; $K_{\text{PZ}}^3 = 0,65$ для случая совместной с разработчиком ПО разработки.

K_{PP}^3 – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ТЗ;

$K_{\text{РП}}^3 = 0,35$ для совместной с разработчиком постановки задачи.

После подстановки принятых значений в (7.1) получаем:

$$\tau_{\text{T3}} = 69 \cdot 0,65 + 69 \cdot 0,35 = 69 \text{ чел.-дни}$$

7.3.2 Расчет трудоемкости выполнения эскизного проекта

Для расчета трудоемкости выполнения эскизного проекта (ЭП) используется формула:

$$\tau_{\text{ЭП}} = T_{\text{РЗ}}^{\vartheta} + T_{\text{РП}}^{\vartheta}, \quad (7.2)$$

где $\tau_{\text{ЭП}}$ – трудоемкость разработки эскизного проекта, чел.-дни;

$T_{\text{РЗ}}^{\vartheta}$ – затраты времени разработчика постановки задачи на разработку ЭП, чел.-дни;

$T_{\text{РП}}^{\vartheta}$ – затраты времени разработчика программного обеспечения на разработку ЭП, чел.-дни.

Для расчета значений $T_{\text{РЗ}}^{\vartheta}$ и $T_{\text{РП}}^{\vartheta}$ могут быть применены следующие формулы:

$$T_{\text{РЗ}}^{\vartheta} = t_{\vartheta} \cdot K_{\text{РЗ}}^{\vartheta}; \quad T_{\text{РП}}^{\vartheta} = t_{\vartheta} \cdot K_{\text{РП}}^{\vartheta},$$

где t_{ϑ} – норма времени на разработку ЭП программного продукта, чел.-дни; $t_{\vartheta} = 94$ чел.-дни, согласно таблице 3, приведенной в [12];

$K_{\text{РЗ}}^{\vartheta}$ – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задачи на стадии ЭП; $K_{\text{РЗ}}^{\vartheta} = 0,7$ для случая совместной с разработчиком ПО разработки.

$K_{\text{РП}}^{\vartheta}$ – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ЭП; $K_{\text{РП}}^{\vartheta} = 0,3$ для совместной с разработчиком постановки задачи разработки ЭП.

После подстановки принятых значений в (7.2) получаем:

$$\tau_{\text{ЭП}} = 94 \cdot 0,7 + 94 \cdot 0,3 = 94 \text{ чел.-дни}$$

7.3.3 Расчет трудоемкости выполнения технического проекта

Трудоемкость разработки технического проекта $\tau_{\text{ТП}}$ зависит от функционального назначения программного продукта, количества разновидностей

форм входной и выходной информации и определяется по формуле:

$$\tau_{\text{тп}} = (t_{\text{РЗ}}^{\text{T}} + t_{\text{РП}}^{\text{T}})K_{\text{В}} \cdot K_{\text{Р}} \quad (7.3)$$

где $t_{\text{РЗ}}^{\text{T}}$, $t_{\text{РП}}^{\text{T}}$ – норма времени, затрачиваемого на разработку ТП разработчиком постановки задач и разработчиком программного обеспечения соответственно, чел.-дни; по таблице 16 в [12] определено, что для принятой группы сложности алгоритма и степени новизны программной продукции нормы времени составляют $t_{\text{РЗ}}^{\text{T}} = 23$ чел.-дни и $t_{\text{РП}}^{\text{T}} = 19$ чел.-дни;

$K_{\text{В}}$ – коэффициент учета вида используемой информации;

$K_{\text{Р}}$ – коэффициент учета режима обработки информации; по таблице 17 приведенной в [12] определено значение $K_{\text{Р}} = 1,45$.

Значение коэффициента $K_{\text{В}}$ определяют из выражения:

$$K_{\text{В}} = \frac{K_{\text{П}} \cdot n_{\text{П}} + K_{\text{НС}} \cdot n_{\text{НС}} + K_{\text{Б}} \cdot n_{\text{Б}}}{n_{\text{П}} + n_{\text{НС}} + n_{\text{Б}}} \quad (7.4)$$

где $K_{\text{П}}$, $K_{\text{НС}}$, $K_{\text{Б}}$ – значения коэффициентов учета вида информации для переменной, нормативно-справочной информации и баз данных соответственно. Согласно таблице 18 методических указаний [12] для группы новизны «Б»: $K_{\text{П}} = 1,20$; $K_{\text{НС}} = 1,08$; $K_{\text{Б}} = 3,12$.

$n_{\text{П}}$, $n_{\text{НС}}$, $n_{\text{Б}}$ – количество наборов данных переменной, нормативно-справочной информации и баз данных соответственно. Для разрабатываемой интегрированной системы $n_{\text{П}} = 2$; $n_{\text{НС}} = 1$; $n_{\text{Б}} = 0$.

После подстановки определенных значений коэффициентов учета вида информации и количества наборов данных в формулу (7.4) получаем:

$$K_{\text{В}} = \frac{1,20 \cdot 2 + 1,08 \cdot 1 + 3,12 \cdot 0}{2 + 1 + 0} = 1,16$$

Для рассчитанного значения $K_{\text{В}}$ по формуле (7.3) получаем значение трудоемкости выполнения технического проекта:

$$\tau_{\text{тп}} = (23 + 19) \cdot 1,16 \cdot 1,45 = 71 \text{ чел.-дни}$$

7.3.4 Расчет трудоемкости выполнения рабочего проекта

Трудоемкость разработки рабочего проекта зависит от функционального назначения программного продукта, количества разновидностей форм

входной и выходной информации, сложности алгоритма функционирования, степени использования готовых программных модулей и уровня алгоритмического языка программирования и определяется по формуле:

$$\tau_{\text{РП}} = K_{\text{K}} \cdot K_{\text{P}} \cdot K_{\text{Я}} \cdot K_3 \cdot K_{\text{ИА}}(t_{\text{РЗ}}^{\text{P}} + t_{\text{РП}}^{\text{P}}) \quad (7.5)$$

где K_{K} – коэффициент учета сложности контроля информации; $K_{\text{K}} = 1,07$ в соответствии с таблицей 19 методических указаний [12].

K_{P} – коэффициент учета режима обработки информации; для рабочего проекта $K_{\text{P}} = 1,52$.

$K_{\text{Я}}$ – коэффициент учета уровня используемого языка программирования; $K_{\text{Я}} = 1$ для языка высокого уровня.

K_3 – коэффициент учета степени использования готовых программных модулей; с учетом применения готового расчетного ядра МКЭ моделирования и большого количества стандартных библиотек $K_3 = 0,6$.

$K_{\text{ИА}}$ – коэффициент учета вида используемой информации и сложности алгоритма;

Значение $K_{\text{ИА}}$ вычисляется по формуле:

$$K_{\text{ИА}} = \frac{K'_{\text{П}} n_{\text{П}} + K'_{\text{НС}} n_{\text{НС}} + K'_{\text{Б}} n_{\text{Б}}}{n_{\text{П}} + n_{\text{НС}} + n_{\text{Б}}} \quad (7.6)$$

где $K'_{\text{П}}$, $K'_{\text{НС}}$, $K'_{\text{Б}}$ – значения коэффициентов учета сложности алгоритма программного продукта и вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно; согласно таблице 22 методических указаний [12] для группы новизны «Б» и группы 1 сложности алгоритма $K'_{\text{П}} = 1,20$, $K'_{\text{НС}} = 0,72$, $K'_{\text{Б}} = 0,60$.

$t_{\text{РЗ}}^{\text{P}}$, $t_{\text{РП}}^{\text{P}}$ – норма времени затраченного на разработку программы на языке высокого уровня разработчиком постановки задач и разработчиком программного обеспечения соответственно, чел.-дни; ранее определено что трудоемкость выполнения технического проекта для разработчика постановки задач $t_{\text{РЗ}}^{\text{T}} = 23$ чел.-дни, тогда по таблице 35а $t_{\text{РЗ}}^{\text{P}} = 13$ чел.-дни, а трудоемкость выполнения технического проекта разработчиком программного обеспечения $t_{\text{РП}}^{\text{T}} = 19$ чел.-дни, следовательно $t_{\text{РП}}^{\text{P}} = 76$ чел.-дни. Под-

ставим табличные значения в формулу (7.6):

$$K_{\text{ИА}} = \frac{1, 20 \cdot 2 + 0, 72 \cdot 1 + 0, 60 \cdot 0}{2 + 1 + 0} = 1, 04$$

Тогда трудоемкость разработки рабочего проекта по формуле (7.5):

$$\tau_{\text{РП}} = 1, 07 \cdot 1, 52 \cdot 1 \cdot 0, 6 \cdot 1, 04(13 + 76) = 90 \text{ чел.-дни}$$

7.3.5 Расчет трудоемкости стадии внедрения

Трудоемкость выполнения стадии “Внедрение” $\tau_{\text{В}}$ может быть рассчитана по формуле

$$\tau_{\text{В}} = (t_{\text{РЗ}}^{\text{В}} + t_{\text{РП}}^{\text{В}})K_{\text{К}} \cdot K_{\text{Р}} \cdot K_{\text{З}} \quad (7.7)$$

где $t_{\text{РЗ}}^{\text{В}}$, $t_{\text{РП}}^{\text{В}}$ – норма времени, затрачиваемого разработчиком постановки задачи и разработчиком программного обеспечения соответственно на выполнение процедур внедрения, чел.-дни; исходя из таблицы 48 $t_{\text{РЗ}}^{\text{В}} = 8$ чел.-дни, $t_{\text{РП}}^{\text{В}} = 24$ чел.-дни.

При подстановке норм времени и коэффициентов учета в формулу (7.7) получаем:

$$\tau_{\text{В}} = (8 + 24)1, 07 \cdot 1, 39 \cdot 0, 6 = 29 \text{ чел.-дни}$$

7.3.6 Расчет суммарной трудоемкости и продолжительности работ

Полученные на предыдущих этапах результаты могут быть сведены в таблицу 7. Общая трудоемкость разработки программной продукции равна сумме величин трудоемкости выполнения отдельных стадий: $\tau_{\text{ПП}} = \tau_{\text{РЗ}} + \tau_{\text{ЭП}} + \tau_{\text{ТП}} + \tau_{\text{РП}} + \tau_{\text{В}} = 69 + 94 + 71 + 90 + 29 = 353$ чел.-дни.

Для целей планирования и контроля используют ленточный график исходя из того что разработку осуществляют небольшой, стабильный по составу коллектив исполнителей. Продолжительность выполнения всех работ по этапам разработки определяют из выражения:

$$T_i = \frac{\tau_i + Q}{n_i} \quad (7.8)$$

где τ_i – трудоемкость i -й работы, чел.-дни;

Q – трудоемкость дополнительных работ, выполняемых исполнителем,

Таблица 7 — Трудоемкости стадий

Стадия разработки	Трудоемкость, чел.-дни	Число исполнителей	Зарплата исполн., руб
Техническое задание	69	1	40 000
Эскизный проект	94	2	42 000
Технический проект	71	2	40 000
Рабочий проект	90	2	40 000
Внедрение	29	1	38 000

чел.-дни;

n_i – количество исполнителей, выполняющих i -ю работу, чел.

Для разрабатываемой интегрированной системы назначим одного исполнителя для подготовки технического задания, двух – для работ по внедрению продукта и выполнения эскизного проекта и трех исполнителей для разработки технического и рабочего проектов. Тогда по формуле (7.8) продолжительность разработки программного продукта:

$$T_{\text{тз}} = \frac{69}{1} = 69 \text{ дни}; \quad T_{\text{эп}} = \frac{94}{2} = 47 \text{ дни}; \quad T_{\text{пп}} = \frac{71}{2} = 36 \text{ дни};$$

$$T_{\text{рп}} = \frac{90}{2} = 45 \text{ дни}; \quad T_{\text{в}} = \frac{29}{1} = 29 \text{ дни}; \quad T_{\Sigma} = 226 \text{ дни};$$

7.4 Определение затрат на разработку интегрированной системы

Затраты на разработку программной продукции могут быть представлены в виде суммы основных статей расходов, таких как амортизационные отчисления, основная заработка плата, дополнительная заработка плата, отчисления на социальное страхование, накладные расходы.

7.4.1 Расчет основной заработной платы

В основную заработную плату включается заработка плата всех исполнителей, непосредственно занятых разработкой программного продукта, с

учетом их должностного оклада и времени участия в разработке. Расчет ведется по формуле:

$$C_{30} = \sum_i \frac{Z_i}{d} \tau_i \quad (7.9)$$

где Z_i – среднемесячный оклад i -го исполнителя, руб.

d – среднее количество рабочих дней в месяце;

τ_i – трудоемкость работ, выполняемых i -м исполнителем, чел.-дни.

$$C_{30} = \frac{40000}{21} \cdot 69 + \frac{42000}{21} \cdot 94 + \frac{40000}{21} \cdot 71 + \frac{40000}{21} \cdot 90 + \frac{38000}{21} \cdot 29 = 678571 \text{ руб.}$$

7.4.2 Расчет дополнительной заработной платы

Дополнительная заработная плата учитывает все выплаты непосредственным исполнителям за время (установленное законодательством), непроработанное на производстве, в том числе: оплата очередных отпусков, компенсация за неиспользованный отпуск и др. Расчет ведется по формуле:

$$C_{3д} = C_{30} \cdot \alpha_{д}$$

где $\alpha_{д}$ – коэффициент отчислений на дополнительную зарплату $\alpha_{д} = 0,2$.

$$C_{3д} = 678571 \cdot 0,2 = 135714 \text{ руб.}$$

7.4.3 Отчисления на социальное страхование

В статье учитываются отчисления в бюджет социального страхования по установленному законодательством тарифу от суммы основной и дополнительной заработной платы:

$$C_{cc} = \alpha_{cc} \cdot (C_{30} + C_{3д})$$

где α_{cc} – коэффициент отчислений на социальное страхование; $\alpha_{cc} = 0,3 + 0,002 = 0,302$.

$$C_{cc} = 0,302 \cdot (678571 + 135714) = 245914 \text{ руб.}$$

7.4.4 Накладные расходы

В статье учитываются затраты на общехозяйственные расходы, непропризвольные расходы и расходы на управление. Накладные расходы определяют в процентном отношении к основной заработной плате:

$$C_{\text{Н}} = \alpha_{\text{Н}} \cdot C_{\text{зо}}$$

где $\alpha_{\text{Н}}$ – коэффициент накладных расходов; $\alpha_{\text{Н}} = 1,8$.

$$C_{\text{Н}} = 1,8 \cdot 678571 = 1221428 \text{ руб.}$$

7.4.5 Амортизационные отчисления на оборудование

В статье учитываются амортизационные отчисления на приобретаемое оборудование. Таблица 8 отображает предполагаемое количество и цену используемого оборудования.

Таблица 8 — Специальное оборудование

№	Наименование	Единица измерения	Кол-во	Цена за единицу, руб.	Сумма, руб.
1	ПЭВМ	шт	3	40000	120000

В случае использования оборудования для разработки нескольких программных продуктов, его стоимость распределяется между всеми продуктами пропорционально времени использования:

$$C_{\text{ко}} = \sum_i \frac{C_{\text{Б}i} \cdot \alpha_i}{100F_{\text{д}}} t_i$$

где $C_{\text{Б}i}$ – балансовая цена i -го вида оборудования, руб.;

α_i – норма годовых амортизационных отчислений для оборудования i -го вида, %; $\alpha_i = 20,0$ % по таблице 50 методических указаний [12];

$F_{\text{д}}$ – действительный годовой фонд времени, ч; $F_{\text{д}} = 2080$ ч при 8-ми часовом рабочем дне и 5-ти дневной рабочей недели;

t_i – время использования i -го вида оборудования для выполнения данной разработки, ч. Затраты на оборудование:

$$C_{\text{ко}} = 3 \cdot \frac{40000 \cdot 20}{100 \cdot 2080} \cdot 226 \cdot 8 = 20861 \text{ руб.}$$

7.4.6 Результаты расчета затрат

По итогам расчетов затрат на разработку программной продукции составлена обобщающая таблица 9:

Таблица 9 — Результаты расчета затрат

№	Наименование	Сметная стоимость, руб.
1	Основная заработка плата	678 571
2	Дополнительная заработка плата	135 714
3	Отчисления на социальное страхование	245 914
4	Накладные расходы	1 221 428
5	Амортизационные отчисления на оборудование	20 861
	Итого	2 302 490

7.5 Выводы

- Суммарная трудоемкость разработки программного продукта составляет 353 чел.-дней.
- Продолжительность выполнения работ всех стадий с учетом принятого числа исполнителей составляет 226 раб. дней.
- Общие расходы на разработку программного продукта составляют 2 302 490 руб.

ГЛАВА 8

ОХРАНА ТРУДА И ЭКОЛОГИЯ

8.1 Введение

При разработке информационной системы необходимо уделить внимание не только техническим и экономическим вопросам, но также проблемам охраны труда и экологии. Во время работы с программным продуктом на пользователя воздействуют как вредные, так и опасные факторы, уровень некоторых из них может превышать допустимые значения. Для обеспечения безопасности работы в данной части проекта проанализирован состав опасных и вредных факторов, причины их возникновения и способы снижения воздействия. Кроме того произведен расчет заземления и предложены мероприятия по охране окружающей среды.

8.2 Опасные факторы

По ГОСТ 12.0.002-80 под опасным производственным фактором понимается производственный фактор, воздействие которого на работающего в определенных условиях приводит к травме, острому отравлению или другому внезапному резкому ухудшению здоровья, или смерти. При работе с компьютером возможно проявление следующих опасных факторов:

1. Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
2. Взрыв трубки кинескопа видеомонитора
3. Возгорание устройств

Определим источник возникновения каждого из указанных факторов и их воздействие на пользователя, а также меры снижения воздействия или защиты.

8.2.1 Воздействие электрического тока

ПЭВМ – это комплекс устройств, работающих от сети переменного тока напряжением 220 вольт частотой 50 Гц. Электрическое напряжение внутри видеомониторов достигает 25000 вольт. В случае прикосновения к открытым частям электрооборудования или нарушении изоляции проводов возникает опасность жизни и здоровью при протекании тока через тело человека.

Исход поражения человека электрическим током зависит от таких факторов как сила тока, время прохождения через организм и частотной характеристики тока. Воздействие электрического тока в общем случае вызывает общие или местные электротравмы [13]. Электрический удар относят к общим электротравмам, его действие выражено в виде возбуждения различных групп мышц, что может привести к судорогам, остановки дыхания и сердечной деятельности. К местным травмам относят ожоги, металлизацию кожи, механические повреждения.

Для предотвращения поражения электрическим током необходимо обеспечить надежную изоляцию токоведущих частей ПЭВМ и сопутствующего оборудования или исключить доступ к ним при помощи защитных корпусов. Для защиты от поражения электрическим током в случае повреждения изоляции должны быть применены следующие меры: защитное заземление, автоматическое отключение питания, двойная (усиленная) изоляция. Необходимо соблюдать правила электропитания используемых устройств: все розетки, предназначенные для подключения ПЭВМ или другого оборудования должны иметь маркировку по напряжению, заземляющие контакты и соответствовать типу, заявленному производителем.

8.2.2 Взрыв кинескопа

Кинескоп видеомонитора ПЭВМ представляет собой электронно-лучевой прибор, имеющий в своем составе баллон, в котором создан глубокий вакуум. За счет внешнего атмосферного давления на баллон создается значительная нагрузка. Таким образом, при механическом нарушении

герметичности баллона, воздух поступает в баллон с большой скоростью, производя эффект взрыва.

Взрыв трубы кинескопа опасен прежде всего образованием большого количества стеклянных осколков, разлетающихся на большой скорости. Попадание осколков в человека вызывает ранения, сопровождающиеся кровотечением. Особенно опасно попадание осколков в область лица.

Для предотвращения взрыва кинескопа следует исключить использование устройств, имеющих механические повреждения по причине возможного развития трещин вакуумного баллона. Также необходимо уделить внимание разработке документов, регламентирующих поведение при работе с ЭЛТ мониторами. Документ должен предостерегать от неаккуратного обращения с такими видеомониторами. Поверхность пола в помещениях эксплуатации ПЭВМ должна быть ровной, без выбоин и не скользкой. По возможности рекомендуется отдавать предпочтение более безопасным современным жидкокристаллическим мониторам, в них не используется электронно-лучевая трубка и, следовательно исключена возможность взрыва.

8.2.3 Возгорание устройств

Опасность возгорания устройств связана с возможностью короткого замыкания внутренних элементов оборудования или цепей электропитания. Кроме того при работе большому нагреву подвержен центральный процессор ПЭВМ, что требует соответствующего охлаждения. Недостаток отвода тепла (например, в результате блокировки вентиляционных отверстий) может привести к перегреву устройства и его возгоранию.

Основные воздействия горения: высокая температура, непосредственное воздействие пламени, газовая среда, задымление. Особое внимание следует уделять риску отравления токсичными продуктами горения. Возгорание отдельных устройств в случае неквалифицированных действий персонала может привести к распространению огня и сильному пожару.

В целях снижения вероятности возгорания необходимо организовать

проведение противопожарного инструктажа для пользователей ПЭВМ, обеспечить надлежащую установку ПЭВМ вдали от отопительных и нагревательных приборов и исключающую риск блокировки систем теплоотвода. Все помещения, где производится эксплуатация ПЭВМ должны быть оборудованы углекислотными огнетушителями типа ОУ-2 или ОУ-5, кондиционерами, приточно-вытяжной вентиляцией и противопожарной сигнализацией и аварийной сетью освещения. В каждом помещении необходимо наличие аптечки первой помощи. Для звукоизоляции и отделки стен и потолков должны применяться несгораемые материалы. Недопустимо нарушение правил обеспечения эвакуационных путей и выходов, их заграждение различными предметами.

8.3 Вредные факторы

Согласно ГОСТ [14] вредным производственным фактором называется производственный фактор, воздействие которого на работающего в определенных условиях может привести к заболеванию, снижению работоспособности и (или) отрицательному влиянию на здоровье потомства. В контексте разрабатываемой системы выделяют возможность воздействия следующих вредных факторов:

Физические:

- 1.1. Повышенный уровень шума
- 1.2. Повышенная запыленность воздуха
- 1.3. Недостаточная освещенность рабочей зоны
- 1.4. Повышенный уровень электромагнитного поля

Психофизиологические:

- 2.1. Умственное напряжение
- 2.2. Напряжение зрительных анализаторов

2.3. Монотонность труда

2.4. Эмоциональные нагрузки

Как и в случае с опасными факторами для вредных факторов будет определен источник и последствия воздействия, а также способы снижения его негативного влияния.

8.3.1 Повышенный уровень шума

Как было описано выше, значительный нагрев центрального процессора ПЭВМ требует эффективного отвода тепла. Для этого как правило применяются системы активного охлаждения. Принцип действия таких систем основан на обдуве вентиляторами тепловых трубок, соприкасающихся с охлаждаемыми устройствами. В результате в корпусе ПВЭМ неотъемлемо находится источник шума — вентилятор. Кроме самих ПЭВМ свой вклад в повышенный уровень шума вносит сопутствующее оборудование, например, печатающие устройства или системы кондиционирования.

Повышенный уровень шума оказывается на состоянии нервной системы человека и ухудшении самочувствия, при длительном действии может быть причиной неврозов. Угнетающее действие на нервную систему проявляется в виде снижения быстроты реакций, замедлении сбора информации и аналитических процессов, в результате приводя к снижению производительности труда и качества работы. В отдельных случаях повышенный уровень шума может вызывать нарушение обмена веществ, сердечно-сосудистые заболевания, изменение скорости дыхания и пульса [13].

Для снижения уровня шума рекомендуется применять следующие меры: использование оборудования, оснащенного вентиляторам пониженной шумности (как правило большего размера и с меньшей частотой вращения) или охлаждаемое пассивно, своевременное техническое обслуживание механических частей оборудования (смазка и очистка от пыли), размещение устройств, уровень шума которых выше допустимого, за пределами рабочих помещений, акустическая обработка помещений. Согласно ГОСТ

12.1.003-83 допустимый уровень звука для помещений пользователей вычислительных машин составляет 50 дБА.

8.3.2 Повышенная запыленность воздуха

При работе компьютер создает электромагнитное поле, источниками которого являются дисплей, процессор, клавиатура. Под действием электромагнитного поля в воздухе образуются заряженные микрочастицы, которые обладают высокой способностью адсорбировать пыль, что приводит к запыленности воздуха.

Нхождение в запыленном воздухе вызывает раздражение кожи и слизистой оболочки дыхательных путей. Продолжительное действие большой концентрации пыли может привести к заболеванием дыхательных путей и легких.

Избежать повышенной запыленности можно при помощи вентиляции помещений и регулярной влажной уборки. Для повышения влажности воздуха, следует применять увлажнители воздуха, это способствует оседанию частиц пыли.

8.3.3 Недостаточная или избыточная освещенность рабочей зоны

В случае неправильно организованного расположения источников света или отклонения их параметров от допустимых значений рабочее место пользователя ЭВМ может быть освещено недостаточно или наоборот избыточно.

При работе с компьютерами большой нагрузке подвергается зрительный анализатор, что может стать причиной повышенной утомляемости глаз и ухудшении зрения.

Снизить вредное воздействие мониторов ПЭВМ на органы зрения позволяет соблюдение режима работы и корректная организация рабочего места пользователя, а также соблюдение требований к видеомонитору по яркости, контрастности и размеру светящейся точки. Длительность чтения информации с экрана не должна превышать 4-х часов, каждый час работы

необходимо отделять перерывом на 5-10 мин, а 2 часа — на 15 мин. Окна в рабочих помещениях должны располагаться с северной и северо-восточной стороны [13]. В качестве источников искусственного освещения должны использоваться люминисцентные лампы белого света (ЛБ) и компактные люминесцентные лампы ДРЛ (дуговая ртутная люминесцентная).

8.3.4 Повышенный уровень электромагнитного поля

Физическое устройство компьютера обуславливает возникновение электромагнитного поля с диапазоном частот от 5 до 400 кГц, уровень которого превышает фоновый. В составе системного блока ПЭВМ имеется большое количество проводников, которые при протекании через них тока являются источником электромагнитного излучения. Одним из наиболее мощных источников электромагнитного поля является видеотерминал, если он относится к типу ЭЛТ мониторов. Основными источниками излучения ЭЛТ видеотерминала являются электронно-лучевая трубка, узлы разверток, импульсный источник питания.

Электромагнитные поля оказывают негативное влияние на организм человека, в том числе на нервную, иммунную и сердечно-сосудистую систему. Механизм воздействия на нервную систему заключается в том, что электромагнитные поля нарушают проницаемость клеточных мембран для ионов кальция, что отрицательным образом сказывается на скорости реакции, памяти и общем психическом состоянии человека. Влияние электромагнитных полей на иммунную систему проявляется в виде нарушения процессов клеточного иммунитета и образования лимфоцитов. Сердечно-сосудистая система человека также подвержена деструктивному воздействию. Отмечается повышение артериального давления, неустойчивости пульса.

Для снижения воздействия электромагнитного излучения на пользователей системы предлагается отказаться от использования ЭЛТ мониторов, а в случае неизбежности применения следует обеспечить организацию рабочих мест таким образом, чтобы пользователи не находились сзади или

сбоку от них, так как эти направления как правило не оборудованы защитными экранами. Соблюдение режима работы также способствует уменьшению вредного эффекта воздействия электромагнитных полей. Без необходимости не следует оставлять компьютер включенными.

8.3.5 Умственное напряжение

Работа компьютерными системами требует анализа большого количества информации, напряжения аналитических ресурсов человека и интенсивной умственной деятельности. Причиной таких нагрузок является сложность предметной области и, следовательно, самих систем.

Черезмерное напряжение интеллектуальных способностей человека приводит к потере работоспособности, ухудшению реакции, психическим расстройствам, нарушению сна.

Снижение умственного напряжения может быть достигнуто благодаря организации перерывов в работе а так же разработке интуитивно понятных графических интерфейсов и инструкций пользователя.

8.3.6 Напряжение зрительных анализаторов

Постоянная концентрация органов зрения на одной точке, расположенной на близком расстоянии вызывает напряжение зрительных анализаторов. Как было отмечено выше, дополнительная нагрузка возникает при нарушениях правил освещения рабочего места.

Регулярные упражнения для глаз, а также соблюдение режима работы за компьютером позволяют уменьшить риск ухудшения зрения. С точки зрения разработчика системы можно предпринять меры для избавления от мелких элементов интерфейса и предоставления возможности настройки размера шрифтов, а также обдуманного выбора используемой цветовой гаммы.

8.3.7 Монотонность труда

Работа с компьютером вынуждает пользователя находиться долгое время в сидячем положении при минимальной физической активности. Деятельность рабочего сводится к однообразным повторяющимся манипуляциям органами управления ПЭВМ — клавиатурой и мышью, при этом зачастую пользователь стремиться минимизировать посторонние действия в избежание потери концентрации.

Действие монотонного труда влияет во-первых на опорно-двигательную систему так как нагрузка долговременно действует на одни и те же мышцы и суставы и во-вторых на центральную нервную систему: монотонные раздражители вызывают дезактивацию мозга, состояние скуки и сонливости.

Периодические перерывы и соблюдение режима работы способствует снижению монотонности труда.

8.3.8 Эмоциональные нагрузки

Эмоциональные нагрузки возникают вследствие дефицита времени, большого объема и плотности информации, боязни утраты данных, повышенной ответственности за возможные ошибки и неожиданных программных сбоев и задержек.

Эмоциональные нагрузки негативно сказываются на сердечно-сосудистой, нервной системе и обмене веществ.

Снижение эмоциональных нагрузок на пользователя может быть достигнуто соблюдением режима работы и регулярными перерывами, а также проектными мерами такими как резервное копирование важных данных, увеличением быстродействия системы или оповещением пользователя посредством графических индикаторов об ожидаемых задержках при выполнении определенных операций.

8.4 Результаты анализа

В данном разделе был произведен анализ вредных и опасных факторов, действующих на пользователя разрабатываемой системы по составу, причине возникновения, влиянию на организм и способах защиты. Сводная информация приведена в таблице 10.

Таблица 10 — Результаты расчета затрат

Опасные	Вредные
Повышенное значение напряжения в электрической цепи	Повышенный уровень шума
Взрыв трубы кинескопа	Повышенная запыленность воздуха
Возгорание устройств	Недостаточная освещенность рабочей зоны Повышенный уровень электромагнитного поля Умственное напряжение Напряжение зрительных анализаторов Монотонность труда Эмоциональные нагрузки

8.5 Расчет заземления

При анализе опасных факторов было выявлено, что для обеспечения электробезопасности следует применять защитное заземление. Произведем расчет заземления по методике описанной в [15]. Напряжение питания сети ПЭВМ и сопутствующего оборудования меньше 1000 В и оно расположено на небольшом расстоянии, поэтому применяется общее заземляющее устройство.

Исходные данные для расчета:

I. Характеристика оборудования: ПЭВМ, в составе которой системный блок мощностью 500 Вт, монитор 30 Вт и дополнительные устройства суммарной мощностью 170 Вт. Рабочее напряжение $U_{\text{раб}} = 220$ В частотой $f = 50$ Гц, $\cos \varphi = 0,8$. Максимальное число одновременно подключенных ПЭВМ равно 6.

II. Электрооборудование расположено в помещении

- III. В качестве естественного заземлителя может быть использована железобетонная свая с квадратным сечением 300×300 мм и длиной $l_c = 3$ м.
- IV. Экспериментальные данные о удельном сопротивлении грунта отсутствуют, но известно, что устройства располагаются во II климатической зоне с чернозёмным грунтом.

8.5.1 Определение предельно допустимого сопротивления заземляющего устройства

В электроустановках напряжением до 1000 В $R_h = 4$ Ом, но если суммарная мощность источников питания, подключенных к сети не превышает 100 кВА, то принимают $R_h = 10$ Ом. Определим суммарную мощность подключенных устройств по формуле:

$$N_\Sigma = \sum_i \frac{N_i}{\cos \varphi} \quad (8.1)$$

где N_Σ – суммарная мощность подключенных устройств, ВА;

N_i – мощность i -го устройства, Вт;

$\cos \varphi$ – коэффициент мощности.

После подстановки исходных данных в формулу (8.1) получаем:

$$N_\Sigma = 6 \cdot \frac{500 + 30 + 170}{0,8} = 5250 \text{ ВА} = 5,25 \text{ кВА} < 100 \text{ кВА}$$

Следовательно определено предельно допустимое сопротивление заземляющего устройства $R_h = 10$ Ом.

8.5.2 Определение сопротивления естественных заземлителей

Правила устройства электроустановок предписывают использовать естественные заземлители. Согласно исходным данным естественным заземлителем может выступать железобетонная свая квадратного сечения со стороной $a = 0,3$ м. Определим сопротивление растеканию тока для данной сваи по формуле:

$$R_c = \frac{0,66\rho}{l_c} \lg \frac{4l_c}{d} \quad (8.2)$$

где R_c – сопротивление растеканию тока сваи, Ом;

ρ – удельное сопротивление грунта, Ом · м;

l_c – длина сваи, м;

d – диаметр сваи, м.

При отсутствии экспериментальных данных удельное сопротивление грунта может быть определено по формуле:

$$\rho = \rho_t \cdot \psi \text{ Ом} \cdot \text{м}$$

где ρ – табличное значение удельного сопротивления, определяемое для каждого вида грунта, Ом · м. Исходными данными определено, что заземление производится в чернозёмную почву, тогда $\rho = 200$ Ом · м;

ψ – коэффициент сезонности, выбирается исходя из климатической зоны. Для заданной II зоны $\psi = 1,6$

Таким образом $\rho = 200 \cdot 1,6 = 320$ Ом · м.

Если свая имеет прямоугольное сечение со сторонами a и b , то диаметр определяется как $d = 2(a + b)/\pi$. При подстановке вычисленных значений в (8.2) получаем:

$$R_c = \frac{0,66 \cdot 320}{3} \lg \frac{4 \cdot 3 \cdot 3,14}{2(0,3 + 0,3)} = 105,40 \text{ Ом}$$

Так как расчетное значение $R_c > R_h$ то необходимо устройство искусственных заземлителей, которые включаются параллельно естественным.

8.5.3 Определение сопротивления искусственных заземлителей

При наличии естественных заземлителей сопротивление искусственных заземлителей не должно быть больше допустимого значения $R_{и.доп}$:

$$R_{и.доп} = \frac{R_e \cdot R_h}{R_e + R_h} = \frac{105,40 \cdot 10}{105,40 + 10} = 11,05 \text{ Ом}$$

Так как напряжение питания ПЭВМ не превышает 1000 В, то применяется сосредоточенное устройство заземления. Примем, что в качестве вертикальных электродов используются стальные прутки диаметром 10 мм и длиной 3 м, расположенные в ряд на глубине $t_0 = 0,7$ м. Сопротивление такого электрода может быть рассчитано по формуле

$$R_e = \frac{0,336\rho}{l} \left(\lg \frac{2l}{d} + \frac{1}{2} \lg \frac{4t + l}{4t - l} \right) \quad (8.3)$$

где R_e – сопротивление электрода, Ом;

l – длина электрода, м;

d – диаметр электрода, м.

t – величина, характеризующая заглубление электрода, м, определяется из соотношения $t = 0,5l + t_0 = 0,5 \cdot 3 + 0,7 = 2,2$ м;

Схема электрода заземления представлена на рисунке 18.

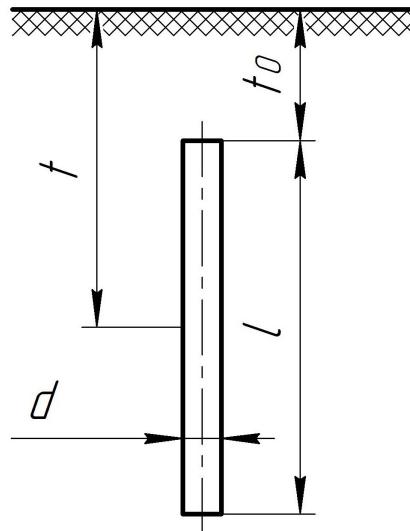


Рисунок 18 – Электрод

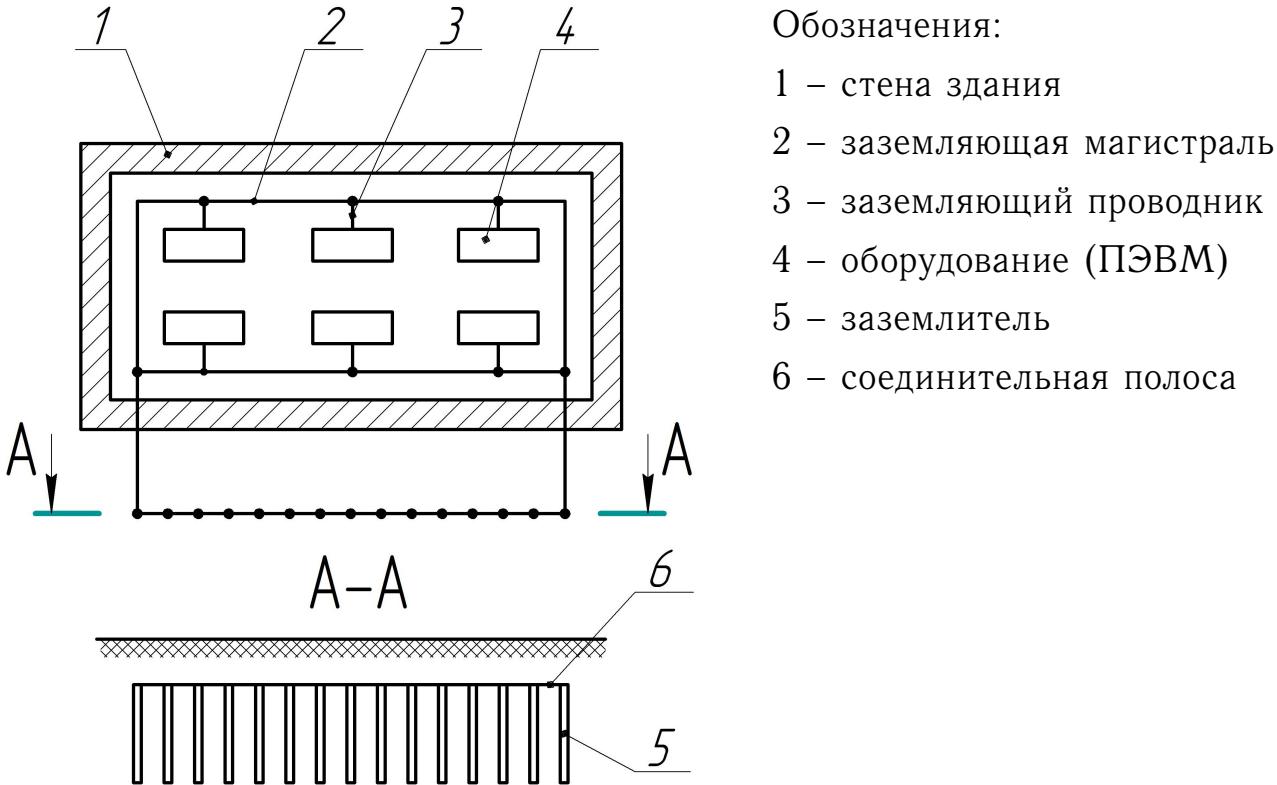


Рисунок 19 – Заземляющее устройство

Подставим величины геометрических параметров стержня в соотношение (8.3):

$$R_{\vartheta} = \frac{0,336 \cdot 320}{3} \left(\lg \frac{2 \cdot 3}{0,01} + \frac{1}{2} \lg \frac{4 \cdot 2,2 + 3}{4 \cdot 2,2 - 3} \right) = 105,10 \text{ Ом}$$

Реальные заземлители представляют собой систему вертикальных электродов, соединенных горизонтальным проводником. В данном случае расположим электроды в ряд с расстоянием $a = 6 \text{ м}$ ($a/l = 2$). Находим произведение коэффициента использования электродов на их количество по формуле $\eta_{\vartheta} \cdot n = R_{\vartheta}/R_{\text{и.доп}} = 105,10/11,05 = 9,51$. Коэффициент использования вертикальных электродов η_{ϑ} учитывает взаимное влияние электрических полей от соседних электродов. По таблице 7.3 приведенной в [15], используя метод интерполяции находим $n = 13,4$. Округляя n в большую сторону до целого получаем количество вертикальных электродов $n = 14$. Схема сосредоточенного заземляющего устройства показана на рисунке 19.

Для соединения вертикальных электродов используем полосовую сталь с сечением $5 \times 10 \text{ мм}$. Длина соединительной полосы может быть определена

Обозначения:

- 1 – стена здания
- 2 – заземляющая магистраль
- 3 – заземляющий проводник
- 4 – оборудование (ПЭВМ)
- 5 – заземлитель
- 6 – соединительная полоса

на по формуле $L = 1,05(n-1)$ $a = 1,05 \cdot (14-1) \cdot 6 = 81,9$ м. Сопротивление растеканию тока горизонтальной полосы:

$$R_{\pi} = \frac{0,336\rho}{L} \lg \frac{2L^2}{ct_0} \quad (8.4)$$

где R_{π} – сопротивление растеканию тока соединительной полосы, Ом;

L – длина соединительной полосы, м;

c – ширина полосы, м;

t_0 – глубина заложения полосы в грунт, м.

Учитывая, что глубина заложения полосы в грунт приблизительно равна глубине верхних концов вертикальных заземлителей получаем:

$$R_{\pi} = \frac{0,336 \cdot 320}{81,9} \lg \frac{2 \cdot 81,9^2}{0,01 \cdot 0,7} = 8,25 \text{ Ом}$$

Результирующее сопротивление искусственного заземлителя растеканию тока определяется по формуле:

$$R_i = \frac{R_{\pi} \cdot R_{\pi}}{R_{\pi} \eta_{\pi} + R_{\pi} \eta_{\pi} \cdot n} \quad (8.5)$$

где R_i – результирующее сопротивление системы вертикальных электродов и соединительной полосы, Ом;

R_{π} – сопротивление вертикального электрода, Ом;

η_{π} – коэффициент использования вертикальных электродов

R_{π} – сопротивление соединительной полосы, Ом;

η_{π} – коэффициент использования соединительной полосы

n – количество вертикальных электродов.

Коэффициент использования вертикального электрода и соединительной полосы выбираем по таблицам 7.3 и 7.4 соответственно (приведены в литературе [15]). Используя метод интерполяции для отсутствующих в таблицах значений получаем $\eta_{\pi} = 0,55$ и $\eta_{\pi} = 0,67$. Таким образом результирующее сопротивление искусственного заземлителя растеканию тока:

$$R_i = \frac{105,10 \cdot 8,25}{105,10 \cdot 0,67 + 8,25 \cdot 0,55 \cdot 14} = 6,45 \text{ Ом}$$

Полученное значение $R_{\text{и}}$ не превышает предельно допустимого $R_{\text{и,доп}} = 11,05$ Ом.

Так как искусственный заземлитель установлен от естественного на большом расстоянии, то пренебрегая взаимным влиянием электрических полей, получаем сопротивление всего заземляющего комплекса:

$$R_3 = \frac{R_e \cdot R_{\text{и}}}{R_e + R_{\text{и}}} = \frac{105,40 \cdot 6,45}{105,40 + 6,45} = 6,08 \text{ Ом}$$

Определенное значение сопротивления искусственного заземлителя меньше установленного нормативного $R_{\text{н}} = 10$ Ом.

8.6 Утилизация печатных плат

Активное использование компьютерной техники в процессе разработки программной системы ставит вопрос о способах утилизации ПЭВМ и их компонент. Наибольший интерес представляет рассмотрение способов утилизации печатных плат – неотъемлемых составляющих ПЭВМ. Во-первых, проблема актуальна с экологической точки зрения, так как печатные платы содержат наиболее опасные для окружающей среды вещества: ртуть, мышьяк, свинец, бериллий, органические загрязнители. Во-вторых, утилизация печатных плат выгодна с экономической точки зрения – в них содержится относительно большое количество ценных металлов: серебро, золото, палладий.

8.6.1 Технологии переработки печатных плат

Печатная плата представляет собой диэлектрическую пластину, служащую платформой для установки и соединения проводниками электронных компонент: микросхем, полупроводниковых элементов, конденсаторов и т.д. Примерный состав печатной платы по материалам, выраженный в процентах от общей массы представлен в таблице 11.

Переработка плат осуществляется в три этапа: предварительная обработка, физическая переработка и химическая переработка. На этапе пред-

Таблица 11 – Состав печатных плат

Стеклополимер	70 %
Медь	16 %
Припой	4 %
Железо, ферриты	3 %
Никель	2 %
Серебро	0,05 %
Золото	0,03 %
Палладий	0,01 %
Прочие (висмут, сурьма, tantal и т.д.)	<0,01 %

варительной обработки осуществляется демонтаж токсичных элементов и элементов, пригодных для повторного использования, разделение и предварительное измельчение.

8.6.2 Физическая переработка

Метод физической переработки заключается в размалывании деталей до требуемых размеров и последующем измельчении отходов до состояния порошка. Полученных порошок при воздействии вихревых токов в сепараторах разделяют на металлическую и неметаллическую фракции. Также для этого применяются и электростатические сепараторы.

Для дальнейшего разделения материалов применяют методы, в основе которых лежит использование различия плотности веществ. Например, метод воздушной сепарации заключается в создании воздушной взвеси частиц. Устройство воздушного сепаратора показано на рисунке 20. Частицы занимают различное положение в зависимости от материала. Более тяжелые частицы имеют предельную скорость осаждения больше, чем скорость воздуха, следовательно, тяжелые частицы перемещаются вниз против воздушного потока, а легкие поднимаются вверх. Таким образом, регулируя скорость истечения воздуха можно добиться выделения из порошка ве-

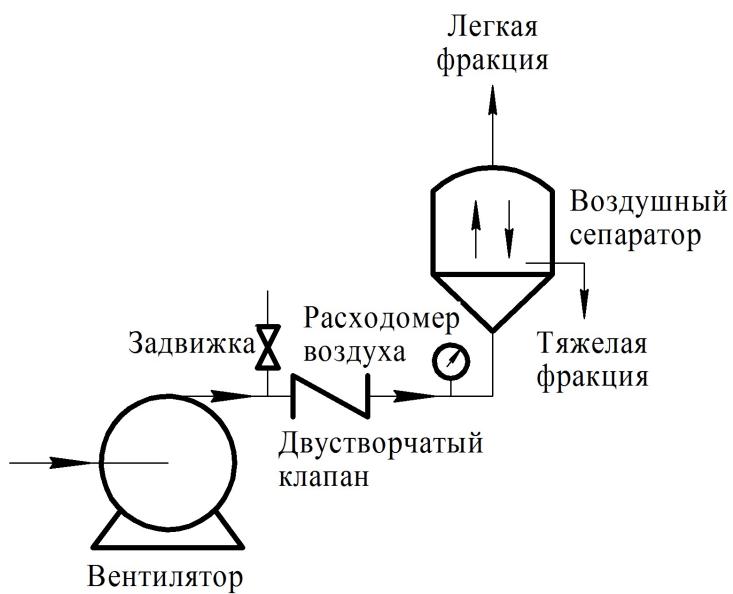


Рисунок 20 – Воздушный сепаратор

ществ различной плотности.

К менее эффективным способам причисляют метод магнитной сепарации. Он позволяет отделить цветные металлы и другие немагнитные материалы от ферромагнитных металлов. Меньшая эффективность метода обусловлена агломерацией частиц, вследствии которой магнит вытягивает некоторую долю немагнитных материалов вместе с магнитными.

8.6.3 Химическая переработка

Для переработки синтетических полимеров широко применяют метод пиролиза. Не исключение и полимеры со стекловолокном, которые входят в состав печатных плат. Пиролиз – это термическое разложение пластмасс при высокой температуре без доступа кислорода. Продуктами пиролиза являются углеродистый обугленный остаток и пиролизный газ. Пиролизный газ при прохождении камеры дожигания теряет свою токсичность при температуре 1400 °C. Углеродистый остаток, содержащий большой процент меди и других металлов называют "черным металлом". Медь выделяют из продуктов пиролиза при помощи электролитического восстановления.

Для извлечения металлических фракций используется гидрометаллур-

гический метод. Он заключается в выщелачивании металлов с применением растворов кислот и щелочей, за которым следует электрографирование желаемых металлов. Суть электрографирования состоит в анодном растворении чернового металла и осаждении на катоде чистого металла. Гидрометаллургический метод экономически эффективен так как позволяет восстанавливать металлы без предварительной термической обработки. К недостаткам метода относят применение ядовитых жидкостей.

Альтернативу описанным способам получения металлов из отходов печатных плат составляют биометаллургические методы. Биометаллургия использует естественные процессы жизнедеятельности микроорганизмов, которые нуждаются в металлах внешней среды для внутриклеточных функций. Существуют некоторые типы микроорганизмов, имеющих тенденцию переносить конкретный металл в определенной среде. Биовыщелачивание применяется в основном для сепарации драгоценных металлов и меди. К преимуществам биометаллургии относят отсутствие вредных выбросов.

8.6.4 Использование неметаллических фракций

Как правило, при переработке печатных плат неметаллические фракции ввиду низкой стоимости не используются, а просто сбрасываются на свалки. Такой подход существенно вредит природе, так как в составе неметаллических фракций сохраняется большой процент тяжелых металлов: свинец, кадмий, бериллий. Чтобы предотвратить загрязнения окружающей среды требуется найти повторное применение такого вида отходов.

Неметаллические фракции могут применяться для получения строительных материалов так как имеют прочную микроструктуру, в составе которой находятся секловолокно. Например, разработана методика использования неметаллических фракций печатных плат в производстве неметаллических пластин, которые могут использоваться для получения композитных плит. Кроме того, неметаллические фракции на бумажной основе можно применять как замену деревянному полу. Немецкая фирма FUBA разработала способ применения стеклополимерных композиций для произ-

водства емкостей и поддонов, стойких к химическому воздействию.

8.6.5 Выводы

В целом, устаревшие и неисправные печатные платы пригодны для переработки и повторного использования. Высокий процент содержания ценных металлов делает их извлечение экономически выгодным. При ответственном подходе к переработке печатных плат есть возможность добиться минимального ущерба окружающей среде, так как даже дешевые неметаллические продукты переработки имеют повторное применение.

СПИСОК ЛИТЕРАТУРЫ

- [1] Метод определения условий механической обработки тонкостенных деталей. / Гаврюшин С.С., Жаргалова А.Д., Лазаренко Г.П. [и др.] // Известия высших учебных заведений. Машиностроение. 2015. № 11. С. 53–60. DOI: 10.18698/0536-1044-2015-11-53-61.
- [2] Шилдт Г. Java. Полное руководство. М.: «Вильямс», 2012. 1104 с.
- [3] А.Г. Косилова, Р.К. Мещеряков. Справочник технолога-машиностроителя. В 2-х т. М.: Машиностроение, 1986. Т. 2. 418 с.
- [4] The RedMonk Programming Language Rankings: June 2015. URL: <http://redmonk.com/sogrady/2015/07/01/language-rankings-6-15>.
- [5] Choosing Static vs. Dynamic Languages for Your Startup - DZone Web Dev. URL: <https://dzone.com/articles/should-your-static-go-static>.
- [6] Bloch J. Effective Java. Addison-Wesley, 2008. 346 p.
- [7] Böck H. The Define Guide to NetBeans Platform 7. Apress, 2012. 558 p.
- [8] Object Management Group. Unified Modeling Language. March 2015. Version 2.5. URL: <http://www.omg.org/spec/UML/2.5>.
- [9] Core J2EE Patterns - Data Access Objects. Sun Microsystems Inc. URL: <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
- [10] Drools Documentation. JBoss. URL: <http://docs.jboss.org/drools/release/6.4.0.Final/drools-docs/html/ch01.html>.
- [11] ГОСТ 19.102-77. ЕСПД. Стадии разработки.
- [12] В.В. Арсеньев, Ю.Б. Сажин. Методические указания к выполнению организационно-экономической части дипломных проектов по созданию программной продукции. Издательство МГТУ, 1994. 52 с.

- [13] Безопасность жизнедеятельности: учебник для вузов / Белов С.В., Девисилов В.А., Ильницкая А.В. [и др.]. 7-е изд., стер. изд. Высш. шк., 2007. 616 с.
- [14] ГОСТ 12.0.002-80. ССБТ. Термины и определения.
- [15] Сборник типовых расчетов для дипломного проектирования / Сивков В.П., Смирнов С.Г., Козыяков А.Ф. [и др.]. Издательство МГТУ, 1978. 80 с.