Оглавление

[Введение. 3](#_Toc233368425)

[1 Аналитический раздел. 4](#_Toc233368426)

[1.1 Постановка задачи. 4](#_Toc233368427)

[1.2 Сложные дискретные системы. 5](#_Toc233368428)

[1.3 Классификация моделей формализации. 8](#_Toc233368429)

[1.3.1 Физические модели. 8](#_Toc233368430)

[1.3.2 Математические модели. 9](#_Toc233368431)

[1.4 Математические модели формализации дискретных систем. 10](#_Toc233368432)

[1.5 Декомпозиция и методы декомпозиции сложных дискретных систем. 12](#_Toc233368433)

[1.6 Определение вероятностного автомата. 16](#_Toc233368434)

[1.7 Сеть вероятностных автоматов и её свойства. 17](#_Toc233368435)

[1.8 Алгоритм декомпозиции вероятностного автомата. 18](#_Toc233368436)

[1.8.1 Разбиение множества. 20](#_Toc233368437)

[1.8.2 СП-разбиение. 21](#_Toc233368438)

[1.8.3 Процедура нахождения всех СП-разбиений. 22](#_Toc233368439)

[1.8.4 Пары разбиений. 22](#_Toc233368440)

[1.8.5 Общая теорема декомпозиции. 23](#_Toc233368441)

[1.8.6 Выбор ортогонального множества разбиений. 28](#_Toc233368442)

[2 Исследовательский раздел. 30](#_Toc233368443)

[2.1 Представление технологического процесса в виде вероятностного автомата. 30](#_Toc233368444)

[2.2 Пример декомпозиции вероятностного конечного автомата. 31](#_Toc233368445)

[2.2.1 Пример декомпозиции не вероятностного конечного автомата. 36](#_Toc233368446)

[2.3 Результаты исследований. 39](#_Toc233368447)

[3 Конструкторский раздел. 40](#_Toc233368448)

[3.1 Общая структура программного продукта. 40](#_Toc233368449)

[3.2 Библиотека для работы с конечными автоматами. 42](#_Toc233368450)

[4 Технологический раздел. 45](#_Toc233368451)

[4.1 Выбор языка программирования и среды разработки. 45](#_Toc233368452)

[4.2 Пользовательский интерфейс. 47](#_Toc233368453)

[4.3 Системные требования. 53](#_Toc233368454)

[4.3.1 Программные требования. 53](#_Toc233368455)

[4.3.2 Аппаратные требования. 53](#_Toc233368456)

[5 Организационно-экономический раздел. 54](#_Toc233368457)

[5.1 Организация и планирование процесса разработки 54](#_Toc233368458)

[5.1.1 Формирование состава выполняемых работ и группировка их по стадиям разработки 54](#_Toc233368459)

[5.2 Расчет трудоемкости выполнения работ 56](#_Toc233368460)

[5.3 Расчет количества исполнителей 61](#_Toc233368461)

[5.4 Календарный план-график разработки ПП 62](#_Toc233368462)

[5.5 Расчёт стоимости программного продукта 63](#_Toc233368463)

[5.6 Расчет экономической эффективности 65](#_Toc233368464)

[5.7 Выводы 66](#_Toc233368465)

[6 Промышленная экология и безопасность 67](#_Toc233368466)

[6.1 Анализ вредных и опасных факторов 67](#_Toc233368467)

[6.1.1 Освещенность 68](#_Toc233368468)

[6.1.2 Электрические и магнитные поля 69](#_Toc233368469)

[6.1.3 Статическое электричество 72](#_Toc233368470)

[6.1.4 Электробезопасность 72](#_Toc233368471)

[6.1.5 Опасность возникновения пожара 73](#_Toc233368472)

[6.1.6 Шум 74](#_Toc233368473)

[6.1.7 Вибрация 75](#_Toc233368474)

[6.1.8 Травматизм 76](#_Toc233368475)

[6.1.9 Микроклимат 77](#_Toc233368476)

[6.1.10 Питьевая вода 79](#_Toc233368477)

[6.2 Расчет освещения 79](#_Toc233368478)

[6.2.1 Расчет площади светопроемов 79](#_Toc233368479)

[6.2.2 Расчет искусственного освещения 82](#_Toc233368480)

[Заключение 88](#_Toc233368481)

[Перспективы развития проекта 89](#_Toc233368482)

[Список литературы. 90](#_Toc233368483)

# Введение.

В настоящее время интерес к моделированию систем постоянно растет. Значительную часть систем составляют именно дискретные системы, поэтому разработка методов их исследования является весьма актуальной задачей.

При исследовании сложных дискретных систем с использованием методов математического моделирования возникает ряд проблем, обусловленных большой размерностью задачи и необходимостью учета множества факторов, присущих системе и влияющих на качество решения задачи. Поэтому одним из важнейших методов исследования таких систем является метод декомпозиции, позволяющий разбивать исследуемую схему на части, проверяя работу каждой части и последовательно добавлять к проверенной части новые фрагменты.

Случайный характер процессов формирования, обработки и передачи данных в сложных дискретных системах обусловливает необходимость применения стохастических моделей, в качестве которых широко используются модели вероятностных автоматов.

При этом встаёт проблема декомпозиции дискретных систем, формализованных в виде P-схем. Поэтому существует необходимость в изучении существующего классического алгоритма декомпозиции конечных автоматов и корректировке его таким образом, чтобы она стал применим к вероятностным автоматам (в своём первоначальном виде он для этого не подходит).

# Аналитический раздел.

## Постановка задачи.

В соответствии с заданием на квалификационную работу необходимо разработать программный продукт, реализующий алгоритм декомпозиции дискретных систем, формализованных в виде вероятностного МП-автомата.

Требования, предъявляемые к программному продукту:

* возможность инициализации вероятностного МП-автомата, включающая задание множеств внутренних состояний, входных и выходных символов;
* декомпозиция заданного вероятностного автомата;
* возможность использования создаваемого программного продукта как части алгоритма анализа сложных систем;
* наличие графического интерфейса пользователя, позволяющего использовать разрабатываемый продукт как самостоятельную программу, решающий задачу декомпозиции вероятностного автомата;
* моделирование работы вероятностной сети, полученной в результате декомпозиции.
* возможность импорта и экспорта исходного вероятностного автомата;

## Сложные дискретные системы.

Рассмотрим системы (устройства) управления, у которых дискретны множества внутренних состояний, входных и выходных сигналов, а также множество моментов времени, в которые поступают входные сигналы, меняются внутренние состояния и выдаются выходные сигналы. Такие системы называют дискретными.

Создание дискретных систем может быть вызвано многими причинами.

Во-первых, принцип действия некоторых элементов, входящих в систему, может быть дискретным. К примеру, в системе управления ракетой имеется импульсная радиолокационная станция (РЛС), измеряющая координаты цели и ракеты. По своему принципу действия она выдает информацию дискретно с частотой следования импульсов станции, поэтому и вся система управления будет дискретной. В качестве другого примера можно указать на системы автоматического управления (САУ), имеющие в своем составе цифровые вычислительные машины (ЦВМ), являющиеся дискретными устройствами.

Во-вторых, в дискретных системах проще реализовать сложные алгоритмы управления. Так, при использовании ЦВМ алгоритм задается в виде программы, сложность которой практически не влияет на конструкцию системы. Смена программы, то есть алгоритма управления, производится без больших затрат времени. В непрерывных же САУ повышение сложности алгоритма управления требует включения в состав системы новых элементов, а замена алгоритма связана с существенным усложнением конструкции.

В-третьих, точность решения алгоритмов управления с помощью дискретных устройств (например, ЦВМ) обычно выше, чем с помощью непрерывных. Это положение требует более подробного объяснения. Дискретная обработка информации за счет импульсного характера сигналов неизбежно приводит к ее потере, так как на интервалах, где импульсы отсутствуют, полезная информация не используется. Поэтому, если для решения одного и того же алгоритма использовать дискретные и непрерывные устройства, то точность последних в идеальном случае будет выше. За счет потери части информации дискретные устройства обладают методической погрешностью, то есть такой, которая зависит от метода обработки. Однако как дискретные, так и непрерывные устройства имеют и другие погрешности - инструментальные, зависящие от неточностей изготовления отдельных элементов, нестабильностей параметров, внутренних шумов и помех. Оказывается, что инструментальные погрешности непрерывных устройств значительно больше, чем устройств дискретных, и сильно растут с усложнением алгоритма обработки. В итоге суммарная погрешность дискретных устройств оказывается меньше инструментальной погрешности непрерывных, что и позволяет говорить о более высокой точности работы дискретных систем [3].

Любое дискретное устройство можно представить в виде набора величин:

, где

Х – множество входных воздействий хi∈Х, i=1…nx

Y – множество выходных характеристик устройства yj∈Y, j=1…ny

V – множество воздействий внешней среды vl∈V, l=1…nv

H – множество (собственных) параметров устройства hk∈H, k=1…nh

В общем случае X, V, H, Y не пересекаемые множества, содержат как детерминированные, так и стохастические составляющие. Процесс функционирования устройства во времени t описывается оператором Fs:

FS может являться функцией, набором логических условий, алгоритмом, таблицей или словесным описанием правил.

Под внутренним состоянием дискретного устройства будем понимать набор значений параметров устройства в заданный момент времени.

Представим дискретное устройство в виде пятерки

, где

XS – множество дискретных входных сигналов,

YS – множество дискретных выходных сигналов,

QS – множество дискретных внутренних состояний устройства,

f – выходная функция,

g – функция переходов

Множества XS, YS и QS связаны со структурой устройства, а функции f и g описывают алгоритм его функционирования. В общем случае, функционирование дискретного устройства можно описать следующим образом:





Где  - упорядоченная последовательность дискретных моментов времени, - значение выходного сигнала в момент времени ti, - значение входного сигнала в момент времени ti, а - состояние дискретного устройства в момент времени ti [2].

Под сложным дискретным устройством будем понимать устройство, обладающее хотя бы один из таких свойств:

* Сложный алгоритм функционирования;
* Наличие случайных факторов влияющих на функционирование устройства [2].

Нетрудно видеть, что описанное устройство можно воспринимать как сложную систему. В дальнейшем, при употреблении термина «система» будет подразумевать «дискретное устройство».

Основным способом изучения сложных систем является моделирование. Рассмотрим основные виды моделей и способы моделирования.

## Классификация моделей формализации.

По степени абстрагирования от оригинала выделяют два больших класса моделей – физические и математические (абстрактные).

### Физические модели.

Физической моделью обычно называют систему, эквивалентную или подобную оригиналу, но возможно имеющую другую физическую природу.

Виды физических моделей представлены в таблице 1.1.

|  |  |
| --- | --- |
| *Вид модели* | *Описание* |
| Натуральные | Это реальные исследуемые системы (макеты, опытные образцы). Имеют полную адекватность (соответствия) с системой оригиналом, но дороги. |
| Масштабная | Это система той же физической природы, что и оригинал, но отличается от него масштабами. Методологической основой масштабного моделирования является теория подобия. При проектировании ВС масштабные модели могут использоваться для анализа вариантов компоновочных решений. |
| Аналоговая | Это система, имеющая физическую природу, отличающуюся от оригинала, но сходные с оригиналом процессы функционирования. Для создания аналоговой модели требуется наличие математического описания изучаемой системы. В качестве аналоговых моделей используются механические, гидравлические, пневматические и электрические системы. Аналоговое моделирование использует при исследовании средства ВТ на уровне логических элементов и электрических цепей, а так же на системном уровне, когда функционирование системы описывается, например, дифференциальными или алгебраическими уравнениями. |

Табл. 1.1 Виды физических моделей.

### Математические модели.

Математические модели представляют собой формализованное представление системы с помощью абстрактного языка, с помощью математических соотношений, отражающих процесс функционирования системы. Для составления математических моделей можно использовать любые математические средства — алгебраическое, дифференциальное, интегральное исчисления, теорию множеств, теорию алгоритмов и т.д.

Виды математических моделей представлены в таблице 1.2.

|  |  |
| --- | --- |
| *Вид модели* | *Описание* |
| Аналитическая | Аналитической моделью называется такое формализованное описание системы, которое позволяет получить зависимость вектора выходных параметров от входных - в явном виде, используя известный математический аппарат. |
| Имитационная | Это совокупность описания системы и внешних воздействий, алгоритмов функционирования системы или правил изменения состояния системы под влиянием внешних и внутренних возмущений. Эти алгоритмы и правила не дают возможности использования имеющихся математических методов аналитического и численного решения, но позволяют имитировать процесс функционирования системы и производить вычисления интересующих характеристик. Имитационные модели могут быть созданы для гораздо более широкого класса объектов и процессов, чем аналитические и численные. |
| Комбинированная | Часть системы формализуется в виде аналитических зависимостей, а часть в виде имитационной модели. Применяется в случае если часть системы можно формализовать аналитически. |

Табл.1.2 Виды математических моделей.

Чаще всего для исследования сложных дискретных устройств используются математические модели, поскольку их разработка существенно дешевле физических и позволяет решать гораздо более широкий спектр задач.

Среди математических моделей рассмотрим имитационные модели, поскольку они позволяют моделировать сложные системы, для которых очень сложно (или даже невозможно) получить аналитическое решение.

Рассмотрим некоторые такие модели.

## Математические модели формализации дискретных систем.

Для проведения моделирования необходимо формализовать устройство. Это можно сделать, используя одну из существующих типовых математических схем (табл. 1.3).

|  |  |
| --- | --- |
| *Тип* | *Описание* |
| D-схемы (непрерывно-детерминированные) | Используются дифференциальные уравнения. Не подходит для дискретных систем. |
| F-схемы (конечные автоматы) | Конечный автомат имеет множество внутренних состояний и входных сигналов, являющихся конечными множествами.  С их помощью описываются узлы и элементы ЭВМ, устройства контроля, регулирования и управления, системы временной и пространственной коммутации в технике обмена информацией. Широта применения F-схем не означает их универсальность. Этот подход непригоден для описания процессов принятия решений, процессов в динамических системах с наличием переходных процессов и стохастических элементов. |
| P-схемы (вероятностные автоматы) | Устройство (система), автоматически изменяющее свое состояние в зависимости от последовательности предыдущих состояний и случайных входных сигналов. Вероятностный автомат используют при моделировании сложных процессов. |
| Q-схемы (системы массового обслуживания) | Динамическая система, предназначенная для эффективного обслуживания случайного потока заявок при ограниченных ресурсах системы. |
| А-схемы | Aгрегативные модели (системы) позволяют описать широкий круг объектов исследования с отображением системного характера этих объектов. Именно при агрегативном описании сложный объект расчленяется на конечное число частей (подсистем), сохраняя при этом связи, обеспечивая взаимодействие частей. |

Табл.1.3 Типовые математические схемы формализации дискретных систем.

Сложность дискретной системы не позволяет сделать однозначный выбор в пользу той или иной математической схемы для построения модели. Для различных дискретных устройств могут подойти различные схемы.

В данной квалификационной работе в качестве схемы формализации системы рассматривается P-схема (далее - вероятностный автомат). Перечислим основные достоинства вероятностных автоматов по сравнению с другими способами формализации моделей:

* Автоматы имеют хорошо определённую семантику, однозначно определяющую поведение этого автомата;
* Имеют наглядное графическое представление;
* Позволяют моделировать широкий класс сложных систем, имеющих стохастический характер;
* Устойчивы к незначительным изменениям моделируемой системы.

Основным недостатком P-схем является невозможность формализации параллельных алгоритмов.

## Декомпозиция и методы декомпозиции сложных дискретных систем.

Моделирование больших систем представляет собой задачу большой размерности, поэтому одним из методов исследования таких систем является метод декомпозиции, позволяющий разбивать исследуемую схему на части, проверяя работу каждой части и последовательно добавлять к проверенной части новые фрагменты.

Известно, что методы декомпозиции делятся по методу формализации сложных дискретных устройств на структурные, объектные и алгоритмические. К структурным методам относятся последовательный, параллельный и последовательно-параллельный метод. Реализации методов декомпозиции различаются по математическому представлению исследуемой системы: в виде конечного автомата, графа или сети. При объектно-ориентированной декомпозиции система разбивается, в соответствии с формализацией ее элементов различными типовыми математическими моделями. При алгоритмической декомпозиции происходит разбиение алгоритма функционирования сложного устройства на модули, где каждый модуль системы выполняет один из этапов общего процесса функционирования.

По точности решения все методы декомпозиции можно разделить на детерминированные и эвристические. Хотя детерминированные методы дают точные результаты, они весьма трудоемки по реализации. Эвристические методы позволяют находить одно локальное решение. Следует также выделить и рандомизированные алгоритмы, ориентированные на случайный поиск, что дает возможность находить серию локальных решений с приближенными результатами.

Анализ точных методов свидетельствует о сложности проблемы оптимального разбиения систем. Их целесообразно применять для систем с малой размерностью (не более 30 компонентов), так как они требуют значительных затрат машинного времени. Приближенные эвристические методы требуют меньших временных затрат в ущерб точности, но их можно применять для систем большой размерности. Известны два варианта последовательного алгоритма разбиения. По первому при формировании очередного блока выделяется подсхема, объем которой заведомо превышает допустимый; блок формируется в результате последовательного исключения компонентов из схемы. По второму, более распространенному, компоненты последовательно включаются в формируемый блок. Последовательно-параллельный алгоритм заключается в одновременном формировании всех блоков разбиения, когда на каждом шаге свободные компоненты распределяются по блокам. Дополнительные возможности дает применение параллельных методов разбиения схем. В частности, предложено использовать дерево свертки для компоновки схем по различным критериям. Такое дерево позволяет частично получить информацию о всей схеме, что отличает данный подход от последовательного, при котором разбиение строится на основе локальной информации. В группе конструктивных алгоритмов последовательные методы требуют меньшего объема оперативной памяти, однако уступают параллельным в быстродействии. Кроме того, их можно применять к схемам различных типов и структур.

При использовании итерационных алгоритмов схему сначала разбивают на определенное число блоков произвольным образом, затем делают перестановки компонентов из одного блока в другой по принятым критериям. Перестановка производится двумя способами: парным и групповым обменом. Для широкого класса схем оказалось эффективным использование алгоритма групповой замены.

Для рандомизированных алгоритмов используют метод решения задачи разбиения, основанный на случайном поиске путем введения случайного шага в процессе разбиения с последующей оценкой его эффективности. Стохастический метод связан с намеренным введением элемен­та случайности в процессе разбиения. Это дает возможность на­ходить серию локальных решений и выбирать наилучший. По адаптивному методу на каждой шаге проверяется выполнение ограничений, устраняются нарушения, и производится локальная, оптимизация разрезания блоков, за которой снова следует проверка выполнения ограничений и т. д.

Описанные методы целесообразно применять при агрегации сложных систем на заданное число подсистем с минимизацией связей между ними.

Сравнивая итерационные алгоритмы с конструктивными, отметим, что структура первых предопределяет их применение к разбиению на фиксированное число блоков, в то время как вторые обычно ориентированы на минимизацию числа блоков. Вместе с тем, итерационные алгоритмы могут успешно применяться в сочетании с конструктивными, образуя группу смешанных методов, при этом возможно применение итерационных методов для улучшения результатов.

Алгоритмы, основанные на последовательных и итерационных методах разбиения графа, описывающего функционировании дискретного устройства на части, являются сложными, трудно поддаются программной реализации и требуют большого объема памяти. Кроме того, качество результата зависит от первоначального разбиения графов и определяется по одному критерию.

Параллельный

Последовательный

Последовательно-параллельный

Метод декомпозиции

Структурный

Алгоритмический

Объектный

Разбиение на подпроцессы

Сильно-связанные подструктуры

КА

Графы

Сети Петри

Рисунок 1.1 Классификация методов декомпозиции.

Как было сказано выше (см. п. ), в данной квалификационной работе в качестве математической модели формализации рассматривается модель вероятностных конечных автоматов. Существующий метод декомпозиции конечных автоматов, основанный на основной теореме декомпозиции автоматов, в своём изначальном виде не применим к вероятностным автоматам. Поэтому для решения поставленной в рамках квалификационной работы задачи необходимо скорректировать и расширить данный алгоритм для возможности применения его к указанному классу систем.

## Определение вероятностного автомата.

Математической моделью дискретного устройства является *абстрактный МП-автомат* (далее просто автомат), определяемый как шестикомпонентный кортеж, или вектор, , у которого:

1. множество состояний (алфавит состояний);
2. множество входных сигналов (входной алфавит);
3. множество выходных сигналов (выходной алфавит);
4. *функция переходов*, реализующая отображение в .
5. *функция выходов*, реализующая отображение в .
6. начальное состояние автомата [1].

*Вероятностным автоматом* называется такой абстрактный автомат, для которого функции и соответственно принимают вид и , где случайная величина , определяющая переход [4].

Рассмотрим пример вероятностного автомата:

1. *.*
2. *.*
3. *.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |
|  | *p* |  | *p* |  | *p* |  | *p* |
|  |  | *0.5* |  |  |  | *0.3* |  |  |
|  | *0.3* |  | *1* |  | *0.4* |  | *0.7* |
|  | *0.2* |  |  |  |  |  |  |
|  |  | *1* |  | *0.5* |  | *1* |  | *0.4* |
|  |  |  | *0.5* |  |  |  |  |

Табл.1.4 Пример задания вероятностного автомата в табличном виде.

В таблице 1.4 на пересечении -ого и -ого столбцов стоят пары переход-вероятность. Сумма всех вероятностей для конкретного и должна . Если , то *вероятность несрабатывания перехода*.

## Сеть вероятностных автоматов и её свойства.

В качестве модели, описывающей совместную работу совокупности вероятностных автоматов, будем использовать понятие вероятностной сети автоматов.

*Сеть автоматов* – это шестёрка

,

где:

1. Z – входной алфавит.
2. , – множество *компонентных автоматов* (КА) *сети*. КА - полуавтомат, его множество состояний, его входной алфавит:

егофункция переходов

1. *W –* выходной алфавит сети.
2. *-* множество *функций соединения* компонентных автоматов сети.
3. *–* множество *входных функций*.
4. *- выходная функция* сети, где случайная величина .

Множества и назовём, соответственно, базисом и структурой сети.

Определённую таким образом вероятностную сеть можно рассматривать как общую модель совместной работы совокупности вероятностных автоматов, поскольку в ней нет ограничений ни на выбор элементов базиса, ни на выбор структуры.

Для сети N можно строить функционально эквивалентный ей автомат , который будем называть результирующим автоматом сети N.

Результирующим автоматом сети назовём автомат

,

у которого:

1. Функция переходов , определяемая следующим образом:

Здесь

1. Функция выходов (в модели Мили), определяемая следующим образом:

В модели Мура

## Алгоритм декомпозиции вероятностного автомата.

Введённые понятия сети автоматов и её результирующего автомата позволяют сформулировать задачи композиции и декомпозиции автоматов. Под задачей композиции автоматов понимается задача нахождения для сети N её результирующего автомата .

Для определения задачи декомпозиции автоматов введём дополнительные понятия.

Автомат называется подавтоматом автомата , если и только если и для любого справедливо

где .

Другими словами, на области определения автомата поведение обоих автоматов совпадает. Таким образом, автомат S «делает столько же, сколько и , и, быть может, несколько больше».

Автоматы *S* и называются *изоморфными*, если существуют три взаимно-однозначных отображения

,

таких, что

и

для любых и , где .

Тройку отображений , и называют *изоморфизмом* автоматов и . Кратко понятие изоморфизма формулируется следующим образом: образ функции равен функции образов. Иначе, изоморфные автоматы идентичны с точностью до обозначений состояний, входных и выходных сигналов.

Автомат назовём *реализацией автомата* (обозначение ), если у автомата *S* существует подавтомат, изоморфный . Таким образом, если автомат *S* реализует автомат , то поведение с точностью до обозначений совпадает с поведением на области определения , так как у автомата *S* должен быть некоторый подавтомат , изоморфный .

Под *задачей декомпозиции автомата S* будем понимать задачу построения сети *N*, такой, что её результирующий автомат реализует заданный автомат *S*, т.е. .

### Разбиение множества.

*Разбиением множества А* называется множество , элементы которого – подмножества , удовлетворяющие следующим условиям:

1. Для любых двух множеств и .
2. *.*

Множества назовём *блоками разбиения* .

Пусть – множество состояний компонентного автомата . Определим разбиение на множестве следующим образом: , если и только если для всех . Таким образом, два состояния и результирующего автомата попадают в один блок , если и только если в их кодах соответственно равны компоненты .

Каждый блок соответствует различным элементам во множестве ( - множество состояний компонентного автомата ), т.е. в столько блоков, сколько различных внутренних состояний имеет система автоматов . Если в , то на можно аналогичным образом задать разбиение (иногда его называют примарным разбиением). Это разбиение, очевидно, определяется следующим образом: , если и только если . Таким образом, в один блок разбиения попадают те состояния результирующего автомата, которые имеют одинаковые *i*-е компоненты. Следовательно, число блоков разбиения равно числу состояний компонентного автомата и между блоками и состояниями имеется взаимно-однозначное соответствие. В связи с этим можно отождествлять состояния с блоками разбиения [1].

### СП-разбиение.

Разбиением на множестве состояний автомата обладает свойством подстановки (является СП-разбиением), если и только если из ( - в одном блоке ) следует, что для всех .

Иначе, под действием любого входного сигнала автомат из состояний, находящихся в одном блоке , переходит в состояния, также находящиеся в одном блоке, т.е. каждый входной сигнал отображает блоки в блоки . Таким образом, для и существует единственный блок , такой, что .

Пусть СП-разбиение на множестве состояний автомата . Тогда образом автомата *S* назовём полуавтомат , у которого , если и только если .

Если и СП-разбиения, то и тоже СП-разбиения.

### Процедура нахождения всех СП-разбиений.

1. Для каждой пары состояний вычисляется наименьшее СП-разбиение , которое отождествляет с (первичные СП-разбиения).
2. Находятся все возможные суммы полученных на первом шаге . Эти суммы образуют вторичные СП-разбиения.

Отождествим два состояния и в одном блоке искомого разбиения . Тогда из определения разбиения с СП следует, что для любого состояния и также должны быть отождествлены , где . Ясно, что если состояние отождествлено с , - с , то состояния и также должны быть отождествлены, поскольку разбиение соответствует эквивалентности, а последняя транзитивна. Процесс повторяется для каждой пары состояний, вошедших в один блок, до тех пор, пока не перестанут отождествляться новые состояния. Построенное таким образом разбиение имеет свойство подстановки и является минимальным разбиением, которое отождествляет состояния и в одном блоке. Чтобы получить другие разбиения, процесс повторяется для каждой пары состояний, т.е. раз, где *M* – число состояний автомата [1].

### Пары разбиений.

Два разбиения определённых на множестве состояний *A* автомата , назовём парой разбиений, если и только если из следует для всех .

Т.е. при работе *S* блоки переводятся в блоки под действием любого входного сигнала, т.е. для каждого и существует единственный блок , такой что .

Если СП-разбиение, то пара разбиений. Также можно показать, что если и две пары разбиений на множестве состояний *A* автомата *S*, то и тоже пары разбиений на множестве *A*.

По аналогии с взаимно-однозначным соответствием между блоками и состояниями , определяет входы в от других автоматов. Т.е. существует взаимно-однозначное соответствие между блоками и состояниями системы автоматов . Как и в любом автомате Мили, следующее состояние (блок разбиения ) определяется его текущим состоянием и входным сигналом (входной сигнал здесь – состояние системы автоматов и буква внешнего входного алфавита ). Таким образом, состояние в любой момент времени определяется состоянием системы автоматов , и , т.е. блоком разбиения и , поэтому пара разбиений, а реализует отображение в .

### Общая теорема декомпозиции.

Пусть некоторое множество разбиений на множестве *A* состояний декомпозируемого автомата .

Теорема. Множеству разбиений можно поставить в соответствие абстрактную сеть автоматов *N*, так чтобы , если и только если

. (1.1)

При этом устанавливается взаимно-однозначное соответствие между разбиениями и компонентными автоматами .

Множество разбиений, удовлетворяющих условию (1.1), будем называть *ортогональным множеством разбиений*.

Таким образом, для декомпозиции автомата необходимо выбрать ортогональное множество разбиений. Способ выбора такого множества будет описан ниже, в соответствующем параграфе.

Поставим в соответствие каждому разбиению функцию , такую, что , т.е. значение функции на паре равно блоку , в котором содержится состояние .

Образуем на множествах A и Z соответственно разбиения и , так что:

1. и находятся в одном блоке разбиения , если и только если для любого справедливо: . Иначе
2. и находятся в одном блоке разбиения , если и только если для любого справедливо: . Иначе

Полученные таким образом пара разбиений, т.е. каждый блок отображается любым входным сигналом в некоторый блок . При этом максимальное разбиение, образующее пару .

Построим сеть , для чего определим все компоненты кортежа *N*. Начнём с входного и выходного алфавитов сети.

1. Полагаем .
2. Полагаем .
3. Построим компонентные автоматы , т.е. определим базис сети.
4. Полагаем .
5. Для определения входного алфавита компонентного автомата воспользуемся построенными разбиениями и . Напомним, что

(1.2)

Здесь и соответственно внутренний и внешний входные алфавиты автомата .

Если на вход функции поступают выходы компонентных автоматов , то пара разбиений, где , так как максимальное разбиение, образующее пару с . Нетрудно также доказать, что . Таким образом, для нахождения автоматов, выходы которых присоединяются ко входу , необходимо найти такое произведение , которое не превосходит , и тогда выходы должны быть соединены со входом .

Определим разбиение следующим образом:

*,* (1.3)

т. е. не входит в это произведение, так как ко входу могут присоединяться выходы других, отличных от , компонентных автоматов.

В автомате полагаем , а определяется согласно равенствам (2).

1. Определим функцию переходов компонентного автомата .

Пусть соответственно блоки разбиений . Если , т. е. (СП-разбиение), то

*.*

Таким образом, значение функции переходов равно блоку разбиения , содержащему . Здесь функция переходов декомпозируемого автомата .

Если же , то

1. Построим функции соединения компонентных автоматов ; иначе (в терминах разбиений) .

Пусть . Образуем множество , такое, что . Таким образом, в попадают только те векторы из , у которых пересечение всех компонентов не пусто. Такое пересечение имеет место, так как компоненты блоки разбиений, т.е. множества.

Функция реализует отображение . Значение определим следующим образом:

т.е. значение функции равно тому блоку разбиения , в который входит пересечение компонентов .

На множестве функция не определена.

1. Определим множество входных функций следующим образом:

т. е. значение функции на равно блоку разбиения , содержащему . Отсюда ясно, что автомат не различают тех букв входного алфавита , которые входят в один блок разбиения .

1. Построим выходную функцию сети , иначе (в терминах разбиений) .

Пусть . Образуем множество , такое, что . Таким образом, в попадают только те векторы из , у которых пересечение всех компонентов не пусто.

Функция реализует отображение . Значение определим следующим образом:

т. е. значение выходной функции сети совпадает со значением функции выхода декомпозируемого автомата на паре , где состояние, попавшее в пересечение компонентов вектора .

На множестве функция не определена.

В работе [ссылка] показано, что построенная таким образом сеть реализует исходный автомат .

Разбиения и однозначно определяется разбиением ; показывает, какие автоматы воздействуют на автомат , а определяет классы неразличимых автоматом букв входного алфавита .

Таким образом, характеристическая тройка автомата . Стоит отметить, что и являются наибольшими разбиениями, причем, чем больше , тем меньше выходов других автоматов воздействует на . Чем больше , тем проще зависимость от внешнего входа . Использование разбиений и при построении является, таким образом, необходимым условием для построения сети наименьшей сложности.

На рисунке 1.2 представлена общая схема алгоритма декомпозиции вероятностного автомата.



Рисунок 1.2 Общая схема алгоритма декомпозиции.

### Выбор ортогонального множества разбиений.

Из конструктивного способа построения сети видно, что структура сети определена в общем случае не однозначно, поскольку неравенство , которое определяет автоматы , влияющие на поведение , может быть выполнено при различных совокупностях разбиений из , где последнее является ортогональным множеством разбиений (т.е. ).

Как было показано выше, от выбора ортогонального множества разбиений зависит структура и состав результирующей сети .

Выбор данного множества должен быть осуществлён до начала декомпозиции автомата. Так как имеется однозначное соответствие между этим выбором и результирующей сетью, то в зависимости от целей декомпозиции можно ввести критерий выбора (оценки) конкретного множества, который позволит получить результат, наиболее полно удовлетворяющий этим целям.

Например, из всех возможных вариантов декомпозиции особый интерес представляют случаи, при которых распределение состояний по подавтоматам сети наиболее равномерно.

Рассмотренный выше критерий даёт количественную оценку каждому множеству ортогональных разбиений (в процентах):

,

где число элементов во множестве ортогональных разбиений, число элементов во множестве состояний исходного автомата, количество элементов в ом блоке, оцениваемого разбиения, количество блоков, оцениваемого разбиения, количество блоков ого разбиения.

Данный критерий даёт тем большую оценку, чем больше разбиение соответствует ему.

# Исследовательский раздел.

## Представление технологического процесса в виде вероятностного автомата.

В реальных условиях многостадийного производства реализация технологического процесса не дает стопроцентную гарантию получения продукции строго заданного качества. Это объясняется влиянием значительного качества технологических величин, часть из которых может не регламентироваться стандартами. Кроме того, сложно выдержать слишком «узкие» допустимые режимы обработки при учёте погрешностей приборов.

То есть, если технология представлена в виде конечных автоматов, то использование любой технологической цепочки, образованной сочетанием строго определенных элементов алфавитов входов и состояний в большинстве случаев не обеспечивает гарантированное попадание в заданные элементы алфавитов выходных величин. Чтобы учесть данную ситуацию необходимо использовать для описания вероятностные автоматы.

Предварительное исследование технологического процесса позволяет проанализировать законы распределения факторов сквозной технологии и определить безусловные вероятности, с которыми каждое из состояний встречается в реальном производстве, и, следовательно, обычно используется. Данные вероятности можно анализировать с помощью гистограмм или диапазонной оценки для отдельных факторов и входных величии, а также и для их совместного появления.

Сочетания алфавитов технологических факторов формируют технологические траектории. Реализация каждой такой траектории позволяет получать определенный уровень свойств и оценивать затраты на её реализацию. Причём, для каждой траектории затраты можно рассчитать с вероятностью близкой к единице. Следовательно, это позволяет использовать детерминированные конечные автоматы для описания влияния технологии на затраты.

Поскольку реализация конкретной технологии не обеспечивает строгое получение конечных свойств определенного качества (т.е. не позволяет получить единственное сочетание алфавитов выходов), то необходимо для каждой технологической цепочки описать свой вероятностный автомат [2].

## Пример декомпозиции вероятностного конечного автомата.

Рассмотрим процесс декомпозиции на конкретном примере вероятностного автомата. Зададим вероятностный автомат в табличном виде (табл. 2.1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 | a5 | a6 |
| z1 | (a1, w2) - 0,5 | (a5, w2) - 0,3 | (a1, w1) - 0,6 | (a6, w1) - 1 | (a1, w3) - 1 | (a2, w2) - 0,6 |
| (a2, w1) - 0,5 | (a1, w2) - 0,2 | (a2, w1) - 0,2 |  |  | (a1, w2) - 0,1 |
|  | (a2, w2) - 0,3 |  |  |  | (a3, w3) - 0,1 |
|  |  |  |  |  | (a4, w3) - 0,1 |
|  |  |  |  |  | (a5, w1) - 0,1 |
| z2 | (a6, w2) - 1 | (a1, w1) - 0,2 | (a5, w3) - 0,8 | (a2, w2) - 0,8 | (a1, w1) - 0,3 | (a6, w2) - 1 |
|  | (a2, w2) - 0,2 | (a2, w3) - 0,1 |  | (a2, w2) - 0,7 |  |
|  | (a3, w3) - 0,2 |  |  |  |  |
|  | (a4, w3) - 0,2 |  |  |  |  |
| z3 | (a6, w1) - 1 | (a1, w1) - 1 | (a5, w1) - 0,9 | (a2, w2) - 0,6 | (a2, w3) - 0,8 | (a5, w3) - 0,7 |
|  |  | (a6, w1) - 0,1 |  | (a3, w3) - 0,2 | (a6, w3) - 0,3 |
| z4 | (a2, w3) - 1 | (a5, w3) - 0,7 | (a1, w1) - 0,5 | (a6, w3) - 1 | (a4, w1) - 0,9 | (a3, w1) - 1 |
|  | (a6, w2) - 0,3 | (a2, w2) - 0,5 |  |  |  |

Табл.2.1 Пример вероятностного автомата, заданного в табличном виде.

Автомат, приведённый в данном примере, содержит в себе как полностью определённые переходы (сумма вероятностей всех возможных исходов из данного состояния при указанном входном символе равна единице) так и частично определённые (сумма вероятностей меньше единицы).

Множество состояний данного автомата , входной алфавит автомата , выходной алфавит автомата .

В качестве множества ортогональных разбиений возьмём множество , где

В результате декомпозиции получаем сеть вероятностных конечных автоматов, представленную на рисунке.

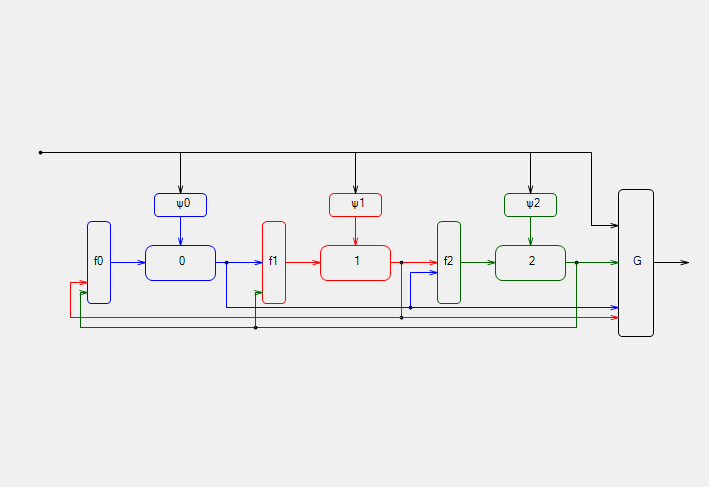


Рисунок 2.1 Сеть вероятностных автоматов.

Стоит отметить, что в результате декомпозиции для подавтоматов сети сформировались следующие состояния:

Для оценки результатов проведём моделирование работы исходного автомата и результирующей сети при следующих условиях:

Начальное состояние : ;

Входная последовательность : ;

Число повторений : **1000**.

Последовательности случайных чисел для сети и автомата **различны**.

Таким образом, в ходе моделирования на вход исследуемого автомата и сети будет подано входных символов.

В результате моделирования работы исходного автомата и результирующей сети были получены результаты, представленные в таблице 2.2.

|  |  |  |
| --- | --- | --- |
|  | Автомат | Сеть |
| Время работы, с | 0,09231 | 0,94221 |
| Число отказов | 175 | 193 |
| Среднее время срабатывания, мс | 0,02308 | 0,23555 |

Табл.2.2 Результаты моделирования исходного автомата и результирующей сети.

Из этих данных можно увидеть, что временные характеристики сети на один порядок ниже временных характеристик исходного автомата. Это связано с вероятностной природой исходного автомата и особенностями программной реализации разработанного алгоритма декомпозиции.

Статистика нахождения автомата и сети в каждом из своих состояний отображена на гистограмме (рисунок 2.2).

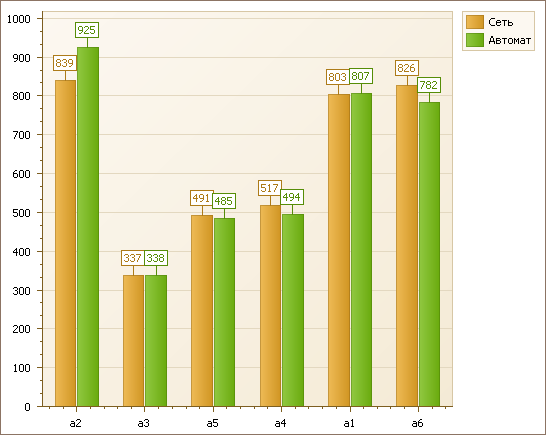


Рисунок 2.2 Гистограмма нахождения автомата и сети в каждом из своих состояний.

Статистика получения выходных символов при работе автомата и сети представлена на гистограмме (рисунок 2.3).

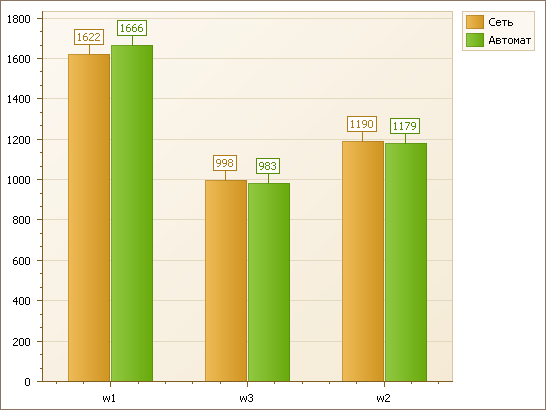


Рисунок 2.3 Гистограмма получения выходных символов при работе автомата и сети.

Рассмотренные гистограммы наглядно демонстрируют, что сеть, полученная в результате декомпозиции, и исходный автомат имеют схожие по характеру распределения. Незначительные различия в столбцах гистограмм обусловлены различными последовательностями случайных чисел, поступающими на вход автомата и сети.

Рассмотрим случай, когда последовательности случайных чисел совпадают. Для этого остальные исходные данные возьмём из предыдущего исследования:

Начальное состояние : ;

Входная последовательность : ;

Статистики распределения состояний и выходных символов представлены на соответствующих диаграммах (рисунки 2.4 и 2.5).

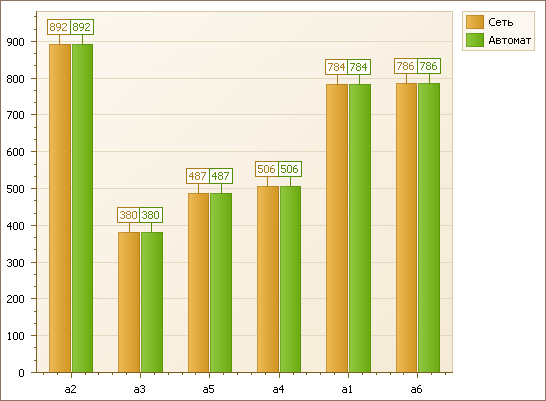
****

Рисунок 2.4 Гистограмма нахождения автомата и сети в каждом из своих состояний.

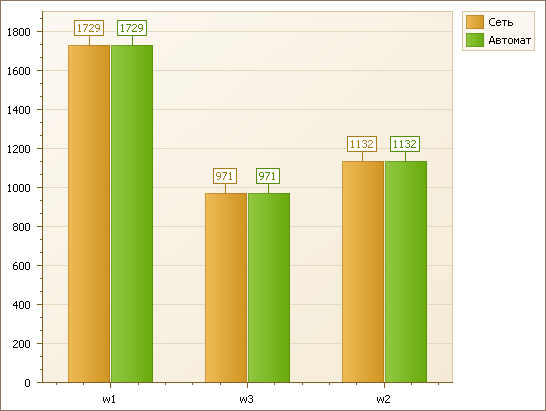
****

Рисунок 2.5 Гистограмма получения выходных символов при работе автомата и сети.

В данном случае распределения полностью совпадают, что позволяет говорить об идентичности поведений исходного автомата и полученной сети. Данный вывод подтверждает корректность разработанного алгоритма декомпозиции.

### Пример декомпозиции не вероятностного конечного автомата.

Рассмотрим результат декомпозиции обычного не вероятностного конечного автомата. Автомат задан в таблице 2.3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 | a5 | a6 |
| z1 | (a2, w2) - 1 | (a5, w2) - 1 | (a1, w1) - 1 | (a6, w1) - 1 | (a1, w3) - 1 | (a2, w2) - 1 |
| z2 | (a4, w2) - 1 | (a1, w1) - 1 | (a5, w3) - 1 | (a2, w2) - 1 | (a2, w2) - 1 | (a3, w2) - 1 |
| z3 | (a6, w1) - 1 | (a1, w1) - 1 | (a5, w1) - 1 | (a2, w2) - 1 | (a2, w3) - 1 | (a5, w3) - 1 |
| z4 | (a2, w3) - 1 | (a5, w3) - 1 | (a1, w1) - 1 | (a6, w3) - 1 | (a4, w1) - 1 | (a3, w1) - 1 |

Табл.2.3 Пример не вероятностного автомата, заданного в табличном виде.

Множество состояний данного автомата , входной алфавит автомата , выходной алфавит автомата .

В качестве множества ортогональных разбиений возьмём множество , где

Для оценки результатов проведём моделирование работы исходного автомата и результирующей сети, полученной при его декомпозиции, при следующих условиях:

Начальное состояние : ;

Входная последовательность : ;

Число повторений : **1000**.

В результате моделирования получены результаты, представленные в таблице.

|  |  |  |
| --- | --- | --- |
|  | Автомат | Сеть |
| Время работы, с | 0,02281 | 0,12386 |
| Среднее время срабатывания, мс | 0,000057 | 0,00311 |

Табл.2.4 Результаты моделирования.

Из данных результатов видно, что общее время работы сети и автомата, имеющих не стохастический характер, снижен практически на 1 порядок

Полученные гистограммы распределений по состояниям и выходным символам, полученные в данном случае, представлены на рисунках 2.6 и 2.7.

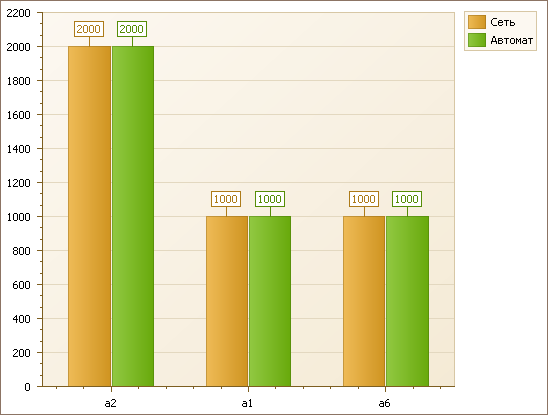


Рисунок 2.6 Гистограмма нахождения автомата и сети на множестве своих состояний.

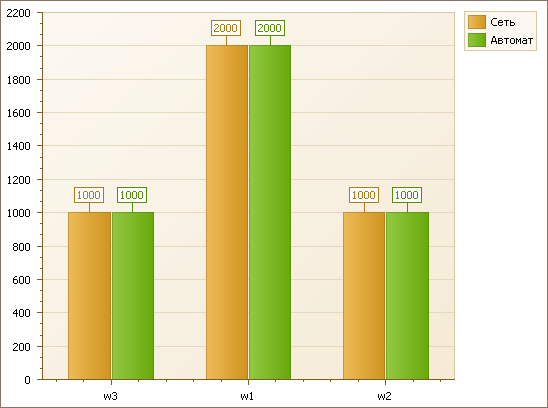


Рисунок 2.7 Гистограмма получения выходных символов при работе автомата и сети.

На представленных гистограммах видно, что распределения идентичны, что объясняется не стохастическим характером исходного автомата.

## Результаты исследований.

Результаты исследований позволяют говорить о том, что получаемая в результате декомпозиции сеть вероятностных автоматов по своему поведению полностью совпадает с поведением исходного автомата.

Проведённые исследования на практике подтверждают предположения о характере временных параметров работы алгоритма для случаев вероятностной и не вероятностной сетей.

При моделировании работы вероятностной сети для каждого входного символа происходит генерирование случайного числа. При этом для всей сети необходимо произвести пересчёт рабочих множеств . При программной реализации это может занять значительное время, что является недостатком исследуемого алгоритма.

# Конструкторский раздел.

## Общая структура программного продукта.

Для того чтобы учесть требование наличия возможности использования создаваемого программного продукта как части алгоритма анализа сложных систем, конечный программный продукт должен представлять собой полностью законченную библиотеку, позволяющую решать установленные в технических требованиях задачи.

Таким образом, программный продукт, получаемый в результате выполнения квалификационной работы, должен состоять из двух основных частей:

* программная библиотека, позволяющая создавать и инициализировать исходный вероятностный автомат, производить его декомпозицию и моделировать работы результирующей сети;
* программное приложение с графическим пользовательским интерфейсом, позволяющее использовать все заявленные функции описанной выше библиотеки.

На рисунке 3.1 представлена общая структура программы.

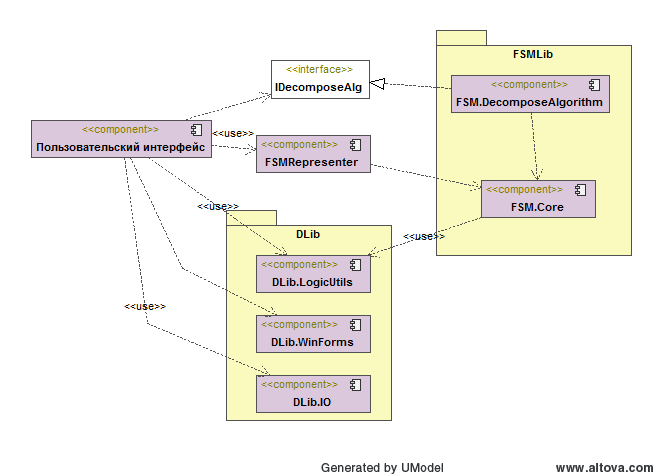


Рисунок 3.1 Общая структура программы.

Программный комплекс состоит из следующих основных модулей:

* **FSM.Core** – содержит все примитивы, связанные с предметной областью теории конечных автоматов, включая определения вероятностного автомата и вероятностной сети автомата;
* **FSM.DecomposeAlgorithm** – содержит реализацию алгоритма декомпозиции вероятностного автомата;
* **IDecomposeAlg** – интерфейс для алгоритма декомпозиции вероятностного автомата, необходим для возможности дальнейшего расширения функционала разрабатываемой библиотеки за счёт увеличения методов декомпозиции;
* **FSMRepresenter** – данный модуль позволяет формировать различное представление данных из предметной области, например в виде таблиц или диаграмм;
* **Пользоватеский интерфейс**;
* **DLib** – утилитарная библиотека, используемая всеми модулями системы; содержит вспомогательные классы общего назначения;

Рассмотрим подробнее модель предметной области.

## Библиотека для работы с конечными автоматами.

При проектировании библиотеки необходимо учитывать тот факт, что в общем случае природа входного и выходного алфавитов неизвестен и определяется только в момент использования разрабатываемой библиотеки. Т.е. в качестве входного алфавита автомат может принимать символы, целые числа, различные структуры, определённые пользователем библиотеки и т.п. Для того, чтобы учесть данный аспект, было принято решение использовать механизм шаблонных типов при проектировании библиотеки.

К достоинствам данного подхода можно отнести использования строгой проверки типов при компилировании программы, что позволяет избежать определённых типов ошибок в процессе написания программного кода. При этом стоит отметить, что использование шаблонной типизации накладывает определённые ограничения на выбор языка программирования (см. технологический раздел).

На рисунке 3.2 представлена принципиальная схема разрабатываемой библиотеки.

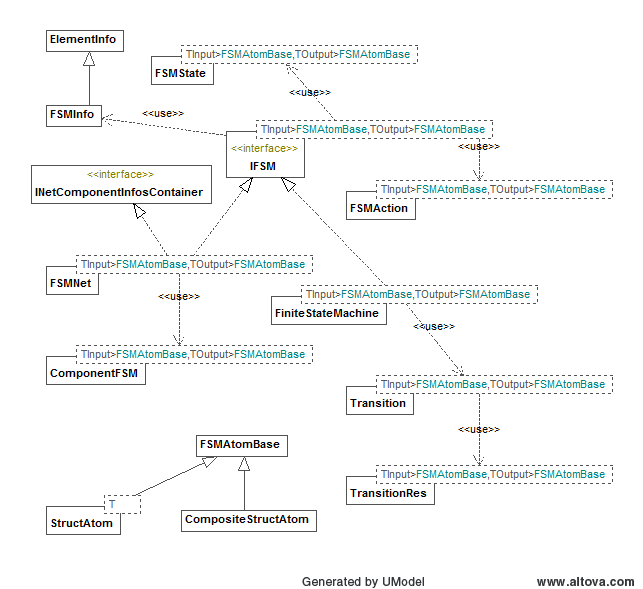


Рисунок 3.2 Диаграмма основных классов программной библиотеки, предназначенной для работы с вероятностными конечными автоматами.

Основные сущности, используемые в библиотеке:

**FSMAtomBase** – базовая сущность для входных и выходных символов автомата. Реализует некоторые утилитарные интерфейсы, а так же логику идентификации, сравнения и копирования символов.

**StructAtom** – обёртка для простых типов входных и выходных символов автомата (целых чисел, строк и т.п.).

**CompositeStructAtom** – обёртка для составных типов входных и выходных символов автомата.

**IFSM** – интерфейс автоматной сущности. Подразумевает наличие имени, входного и выходного алфавитов, множества состояний, способности обрабатывать входные символы. Для определения сущностей данного типа необходимо указать шаблонные типы входных и выходных типов.

**FiniteStateMachine** – вероятностный конечный автомат, реализует интерфейс **IFSM**.

**FSMNet** – сеть вероятностных автоматов (подавтоматов). Реализует интерфейс **IFSM**. Содержит коллекцию подавтоматов.

**ComponentFSM** – подавтомат сети вероятностных автоматов.

**FSMState** – сущность состояния автомата, содержит информацию обо всех переходах из данного состояния.

**FSMAction** – сущность действия, совершаемого при срабатывании перехода.

**Transition** – сущность перехода из заданного состояния при воздействии определённого входного символа. Содержит коллекцию возможных исходов (**TransitionRes**).

**TransitionRes** – сущность исхода перехода (**Transition**). Содержит результирующее состояние, выходной символ и вероятность данного исхода.

# Технологический раздел.

## Выбор языка программирования и среды разработки.

Рассматриваемый программный комплекс реализован с использованием принципов объектно-ориентированного программирования. Для организации и проектирования объектной модели программы использовались шаблоны («паттерны») объектно-ориентированного проектирования. Такой подход позволяет создавать гибкий дизайн приложения, способный учитывать всевозможные расширения функциональности программы и приводит к оптимальному уровню абстракции данных.

В настоящее время существует множество языков и сред программирования, многие из которых обладают достаточно высокой эффективностью, удобством и простотой в использовании.

Из языков программирования, которые теоретически подходят для реализации поставленной задачи, автору известны C, C++, Object Pascal, C#, Managed C++, Visual Basic.

Для данного курсового проекта была выбрана платформа **.NET** и среда программирования **MS Visual Studio 2008.**

Выбор данной платформы обусловлен большим выбором стандартных библиотек, позволяющих разрабатывать программные приложения с наибольшей эффективностью при минимальных затратах времени. В качестве основного языка программирования выбор был остановлен на **С#**.

Из возможных версий платформы **Microsoft .NET Framework** была выбрана наиболее актуальная (из стабильных на момент начала работы над дипломным проектом) версия 3.5, т.к. в данной версии была введена поддержка синтаксиса языка C# 3.0.

C# - один из самых распространённых и популярных современных языков объектно-ориентированного программирования. Данный язык был выбран в силу того, что он имеет широкие возможности по написанию сложных функциональных программных комплексов, использующих различные ресурсы и имеющих гибкий пользовательский интерфейс. С# позволяет в короткие сроки создавать развитые иерархии взаимодействующих классов в каждом логическом слое приложения и связывать их функциональность со слоем отображения, реализую событийную модель взаимодействия компонентов. Кроме того, существуют огромнейшие библиотеки классов (.NET Framework, последняя версия - 3.5), написанные на этом языке, где можно найти практически всё, что требуется для работы над проектом любой сложности. Также C# предоставляет широкие возможности по использованию механизмов кодогенерации в самых различных её проявлениях (в третьей версии появились автосвойства, анонимные делегаты, лямбда-функции и прочее), работе с хранилищами данных.

Выбор среды разработки обусловлен следующими причинами:

* 1. данная среда является самой современной из доступных нам на сегодняшний день и позволяет использовать весь спектр технологий и программных библиотек;
  2. данная среда программирования является промышленным стандартом разработки приложений;
  3. среда разработки MS Visual Studio 2008 предоставляет широкие возможности при написании кода, его редактировании, а также позволяет эффективно отлаживать программы с использованием различных инструментов, что позволяет качественно и быстро устранять возникающие ошибки;
  4. данная среда предоставляет удобную систему отладки кода программ и множество мастеров по созданию функциональных блоков программ (компонентов, визуальных элементов, дизайнеров и прочее);
  5. одно из главных достоинств среды разработки – высокая надежность в работе с проектами.
  6. наличие бесплатной студенческой версии.

## Пользовательский интерфейс.

Главное окно программы представлено на рисунке 4.1.

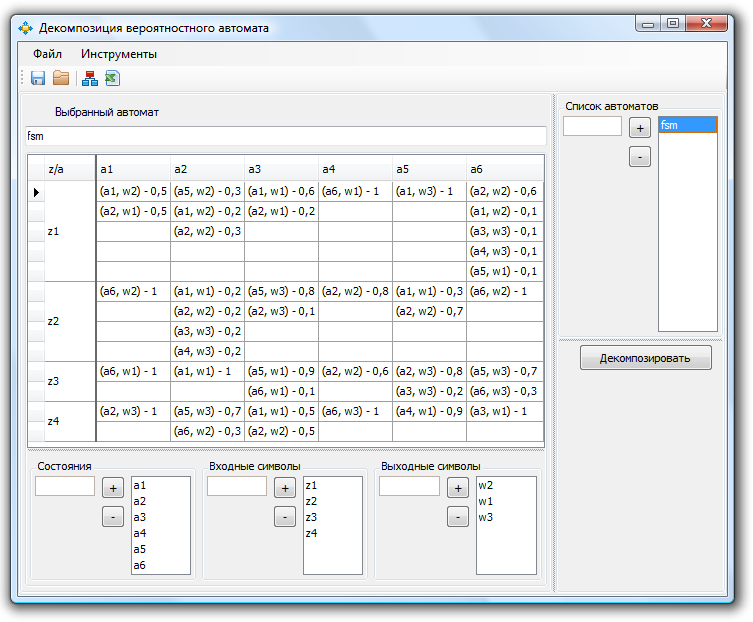


Рисунок 4.1 Главное окно программы.

Основную часть окна занимает табличное представление выбранного вероятностного автомата. В столбцах таблицы находятся состояния автомата, а в строках – его входной алфавит. На пересечениях строк и столбцов расположены возможные исходы для данного состояния при условии прихода заданного символа. Каждый исход представлен в виде записи (a, w) – p, где a – это конечное состояние данного исхода, w – выходной символ, а p – вероятность исхода.

Список всех загруженных автоматов находится в правой части окна. Данный список снабжён специальными кнопками, находящимися рядом с ним и позволяющими добавлять и удалять автоматы.

В нижней части окна находятся подобные списки для редактирования множества состояний текущего автомата, его входного и выходного алфавитов.

Для редактирования переходов автомата необходимо выделить любую ячейку и нажать соответствующую кнопку «…». Это приведёт к появлению окна редактирования множества переходов для заданных состояния a и входного символа z (рисунок 4.2).

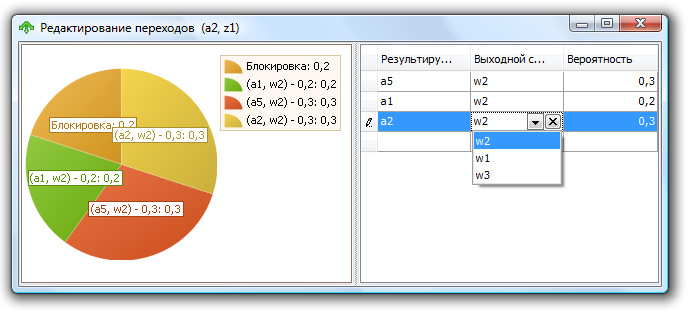


Рисунок 4.2 Окно редактирования переходов.

В правой части окна расположено табличное представление возможных переходов, позволяющее производить их редактирование. В левой части находится круговая диаграмма, представляющая распределение вероятностей различных исходов.

Для редактирования исходов можно использовать встроенные в таблицу кнопки редактирования или же отдельное окно (рисунок 4.3).

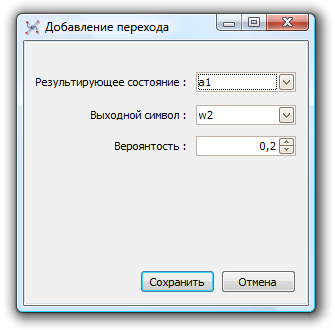


Рисунок 4.3 Окно редактирование перехода.

После задания исходного автомата можно произвести его декомпозицию. Для этого необходимо использовать кнопку **«Декомпозиция»** на форме главного окна.

Как следует из алгоритма, до начала декомпозиции необходимо выбрать множество ортогональных разбиений. Для этого служат окна редактирования разбиений (рисунки 4.4 и 4.5).

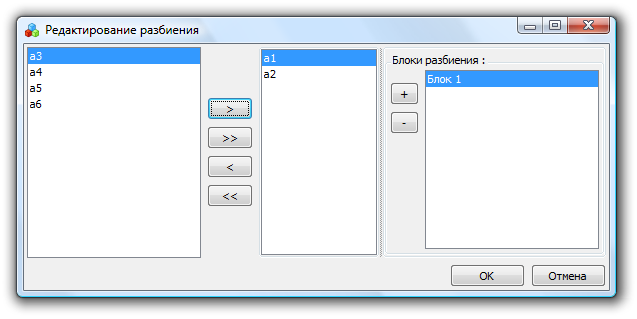


Рисунок 4.4 Окно редактирования разбиения.

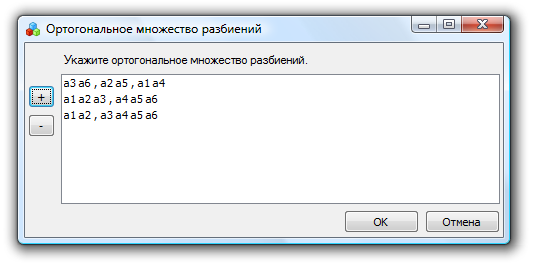


Рисунок 4.5 Окно редактирования множества ортогональных разбиений.

В ходе создания разбиений программа автоматически предлагает завершения множества ортогональных разбиений (рисунок 4.6).

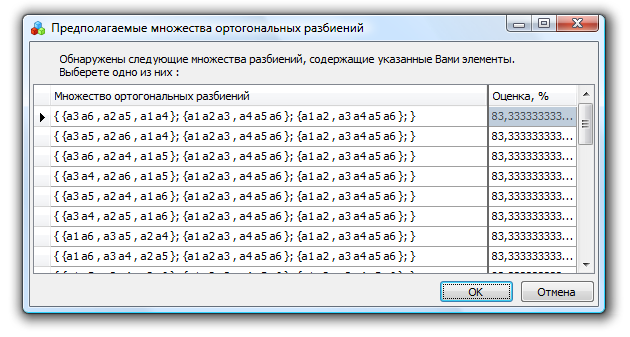


Рисунок 4.6 Окно редактирования разбиения.

После выбора множества ортогональных разбиений происходит декомпозиция автомата. Результаты декомпозиции отображаются в специальном окне (рисунок 4.7).

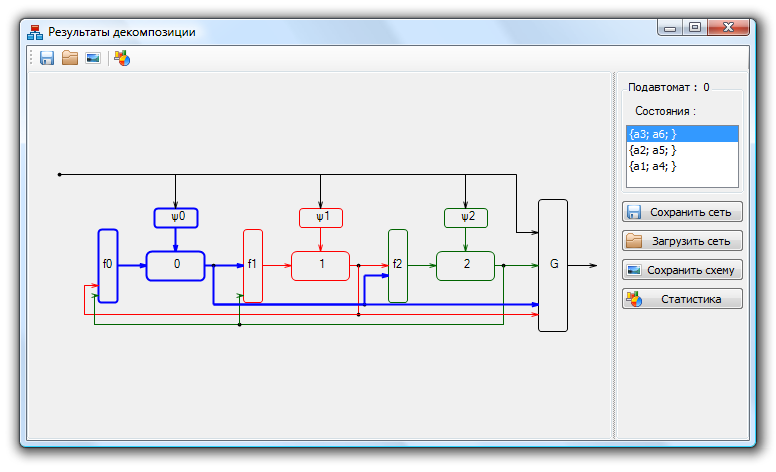


Рисунок 4.7 Окно результатов декомпозиции.

В данном окне основное место занимает схема, полученной в результате декомпозиции сети. В правой части располагается список состояний выделенного элемента сети.

Из данного окна можно вызвать окно моделирования (рисунок 4.8).

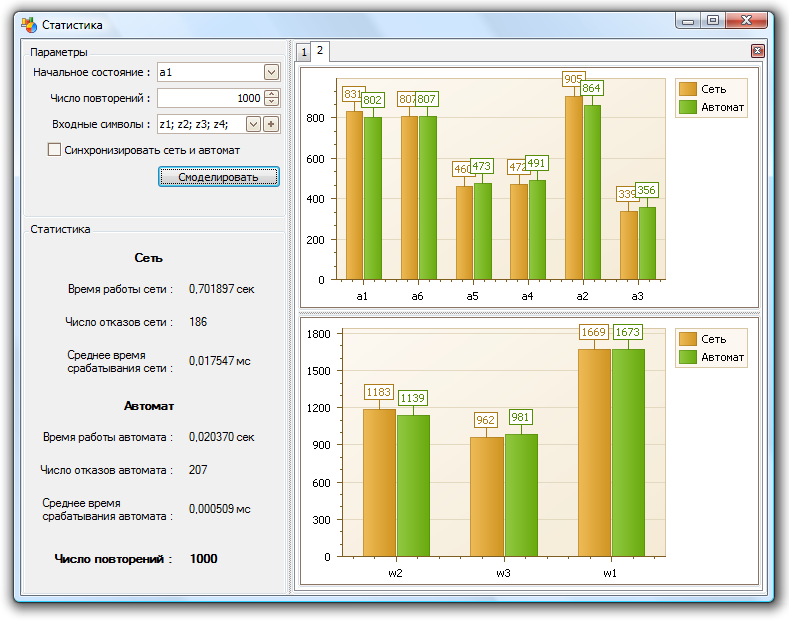


Рисунок 4.8 Окно моделирования.

Данное окно позволяет производить моделирование исходного автомата и сети, полученной в результате декомпозиции.

Результаты всех проведённых экспериментов оформляются в виде вкладок в правой области окна. В верхней части представлена гистограмма нахождения автомата и сети в каждом из своих состояний за всё время эксперимента. В нижней части представлена гистограмма получения выходных символов при работе автомата и сети.

Для проведения моделирования необходимо задать начальное состояние, входную последовательность и количество её повторений.

Входная последовательность задаётся, используя специальный редактор (рисунок 4.9).

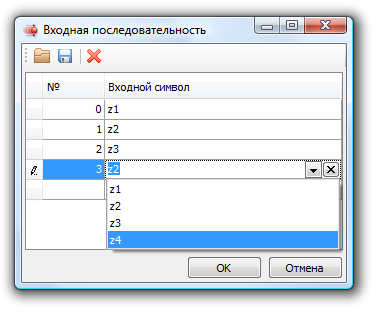


Рисунок 4.9 Окно редактирования входной последовательности.

## Системные требования.

### Программные требования.

* Microsoft .NET Framework 3.5.

### Аппаратные требования.

* Процессор с частотой 1.5 ГГц.
* Интегрированная видеокарта.
* 1024 Мб RAM

# Организационно-экономический раздел.

## Организация и планирование процесса разработки

При использовании традиционного подхода, организация и планирование процесса разработки программного продукта или программного комплекса предусматривает выполнение следующих работ:

* формирование состава выполняемых работ и группировка их по стадиям разработки;
* расчет трудоемкости выполнения работ;
* установление профессионального состава и расчет количества исполнителей;
* определение продолжительности выполнения отдельных этапов разработки;
* построение календарного графика выполнения разработки;
* контроль выполнения календарного графика.

### Формирование состава выполняемых работ и группировка их по стадиям разработки

Разработку программного продукта можно разделить на следующие стадии:

**Техническое задание.** Постановка задач. Определение состава пакета прикладных программ, состава и структуры информационной базы. Выбор языков программирования. 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. Консультации разработчиков постановки задач. Согласование и утверждение эскизного проекта.

**Технический проект.** Разработка алгоритмов решения задач. Разработка пояснительной записки. Согласование и утверждение технического проекта. Разработка структуры программы. Разработка программной документации и передача ее для включения в технический проект. Уточнение структуры, анализ и определение формы представления входных и выходных данных. Выбор конфигурации технических средств.

**Рабочий проект.** Комплексная отладка задач и сдача в опытную эксплуатацию. Разработка проектной документации. Программирование и отладка программ. Описание контрольного примера. Разработка программной 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.

Планирование длительности этапов и содержания проекта осуществляется в соответствии с ЕСПД ГОСТ 34.603-92 и распределяет работы по этапам, как показано в таблице 5.1.

| **Основные стадии** | **№** | **Содержание работы** |
| --- | --- | --- |
| 1.Техническое задание | 1 | Постановка задачи |
| 2 | Выбор средств разработки и реализации |
| 2.Эскизный проект | 3 | Разработка структурной схемы системы |
| 4 | Разработка структур данных |
| 5 | Разработка алгоритмов решения частных задач |
| 3. Техно-рабочий проект | 6 | Реализация структур хранения данных |
| 7 | Реализация алгоритмов решения частных задач |
| 8 | Разработка пользовательского интерфейса |
| 9 | Реализация пользовательского интерфейса |
| 10 | Отладка всего комплекса |
| 11 | Исправление ошибок и недочетов |
| 12 | Разработка документации к системе |
| 13 | Итоговое тестирование системы |
| 4. Внедрение | 14 | Установка и настройка ПП |

Табл.5.1 Распределение работ проекта по этапам.

## Расчет трудоемкости выполнения работ

Трудоемкость разработки программной продукции зависит от ряда факторов, основными из которых являются следующие:

* степень новизны разрабатываемого программного комплекса,
* сложность алгоритма его функционирования,
* объем используемой информации, вид ее представления и способ обработки,
* уровень используемого алгоритмического языка программирования (Чем выше уровень языка, тем меньше трудоемкость).

|  |  |
| --- | --- |
| Функциональное назначение ПП | Управление НИР и САР, научно-технической информацией, документопотоком, охраной природы и окружающей среды. |
| Степень новизны разрабатываемого проекта | Группа новизны **В** - разработка программной продукции, имеющей аналоги. |
| Степень сложности алгоритма функционирования | **3** группа сложности - программная продукция, реализующая алгоритмы стандартных методов решения задач. |
| По виду представления исходной информации | Группа **12** - исходная информация представлена в форме документов, имеющих одинаковый формат и структуру, требуется форматный контроль информации. |
| Структура выходных документов | Группа **22** - требуется вывод на печать одинаковых документов, вывод информационных массивов на машинные носители. |

Табл.5.2 Исходные данные.

Трудоемкость разработки программной продукции τПП может быть определена как сумма величин трудоемкости выполнения отдельных стадий разработки ПП из выражения:

τПП = τТЗ + τЭП + τТП + τРП + τВ, (5.1), где

τТЗ – трудоемкость разработки технического задания на создание ПП;

τЭП – трудоемкость разработки эскизного проекта ПП;

τТП – трудоемкость разработки технического проекта ПП;

τРП – трудоемкость разработки рабочего проекта ПП;

τВ - трудоемкость внедрения разработанного ПП.

Трудоемкость разработки технического задания рассчитывается по формуле:

τТЗ = TРЗЗ + TРПЗ, (5.2), где

TРЗЗ – затраты времени разработчика постановки задач на разработку ТЗ, чел.-дни;

TРПЗ – затраты времени разработчика программного обеспечения на разработку ТЗ, чел.-дни.

Значения величин ТРЗЗ и ТРПЗ рассчитываются по формулам

TРЗЗ = tЗ ⋅ KРЗЗ; (5.3)

TРПЗ = tЗ ⋅ KРПЗ, (5.4), где

tЗ – норма времени на разработку ТЗ на программный продукт в зависимости от функционального назначения и степени новизны разрабатываемого ПП, чел.-дни;

KРЗ3 – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки на стадии ТЗ;

KРП3 – коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ТЗ.

tЗ = 24 [чел.-дн.]

KРЗЗ = 0,65

KРПЗ = 0,35

τТЗ = 24 ⋅ ( 0,65 + 0,35 ) = 24 [чел.-дн.]

Аналогично рассчитывается трудоемкость эскизного проекта ПП τЭП:

τЭП = TРЗЭ + TРПЭ = 50 (5.5)

ТРЗЭ = tЭ ⋅ KРЗЭ = 50 ⋅ 0,75

ТРПЭ = tЭ ⋅ KРЗЭ = 50 ⋅ 0,25

τЭП = 50 ⋅ ( 0,60 + 0,40 ) = 50 [чел.-дн.]

Трудоемкость разработки технического проекта τТП зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации и определяется как сумма времени, затраченного разработчиком постановки задач и разработчиком программного обеспечения, т.е.

τТП = (tРЗТ + tРПТ) ⋅ KВ ⋅ KР, (5.6), где

tРЗТ, tРПТ – норма времени, затрачиваемого на разработку технического проекта (ТП) разработчиком постановки задач и разработчиком программного обеспечения соответственно, чел.-дни;

KВ – коэффициент учета вида используемой информации;

KР – коэффициент учета режима обработки информации.

Значение коэффициента KВ определяется из выражения:

KВ = (KП ⋅ nП + KНС ⋅ nНС + KБ ⋅ nБ) / (nП + nНС + nБ) (5.7), где

KП, KНС, KБ – значения коэффициентов учета вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно;

nП, nНС, nБ – количество наборов данных переменной, нормативно-справочной информации и баз данных соответственно.

KР = 1,36 (согласно таблице о значении коэффициента учета режима обработки информации)

KП = 1,00; KНС = 0,72; KБ = 2,08

KВ = 1,08

tРЗТ = 38; tРПТ = 12

τТП = ( 38 + 12 ) ⋅ 1,08 ⋅ 1,36 = 73 [чел.-дн.]

Трудоемкость разработки рабочего проекта τРП зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации, сложности алгоритма функционирования, сложности контроля информации, степени использования готовых программных модулей, уровня алгоритмического языка программирования и определяется по формуле:

τРП = KК ⋅ KР ⋅ KЯ ⋅ KЗ ⋅ KИА⋅ (tРЗР + tРПР), (5.8), где

KК – коэффициент учета сложности контроля информации;

KЯ – коэффициент учета уровня используемого алгоритмического языка программирования;

KЗ – коэффициент учета степени использования готовых программных модулей;

KИА – коэффициент учета вида используемой информации и сложности алгоритма ПП;

tРЗР, tРПР – норма времени, затраченного на разработку РП на алгоритмическом языке высокого уровня разработчиком постановки задач и разработчиком программного обеспечения соответственно, чел.-дни.

Значение коэффициента KИА определяется из выражения

KИА = (KП′ ⋅ nП + KНС′ ⋅ nНС + KБ′ ⋅ nБ) / (nП + nНС + nБ), (5.9), где

KП′, KНС′, KБ′ – значения коэффициентов учета сложности алгоритма ПП и вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно.

KК = 1

KР = 1,44 (для рабочего проекта)

KЯ = 1

KЗ = 0,5

tРЗР = 10 [чел.-дн.]

tРПР = 54 [чел.-дн.]

KП′ = 1; KНС′ = 0,48; KБ′ = 0,4

KИА = 0,51

τРП = (10 + 54) ⋅ 1 ⋅ 1.44⋅ 1 ⋅ 0.5 ⋅ 0.51 = 24 [чел.-дн.]

Так как при разработке ПП стадии «Технический проект» и «Рабочий проект» объединены в стадию «Техно-рабочий проект», то трудоемкость ее выполнения τТРП определяется по формуле:

τТРП = 0,85 ⋅ τТП + τРП (5.10)

τТРП = 0,85 ⋅ 73 + 24 = 86 [чел.-дн.]

Трудоемкость выполнения стадии внедрения τВ может быть рассчитана по формуле:

τВ = (tРЗВ + tРПВ) ⋅ KК ⋅ KР ⋅ KЗ, (5.11), где

tРЗВ, tРПВ – норма времени, затрачиваемого разработчиком постановки задач и разработчиком программного обеспечения соответственно на выполнение процедур внедрения ПП, чел.-дни.

KР = 1,26

tРЗВ = 11 [чел.-дн.]

tРПВ = 12 [чел.-дн.]

τВ = (11 + 12) ⋅ 1 ⋅ 1,26 ⋅ 0,5 = 14 [чел.-дн.]

Подставляя полученные данные в (5.1), получим:

τПП = 24 + 50 + 86 + 14 = 174 [чел.-дн.]

| **Этап** | **Трудо-емкость этапа** | **№ работы** | **Содержание работы** | **Трудоемкость, чел-дн.** |
| --- | --- | --- | --- | --- |
| 1 (ТЗ) | 24 | 1 | Постановка задачи | 20 |
| 2 | Выбор средств разработки и реализации | 4 |
| 2 (ЭП) | 50 | 3 | Разработка структурной схемы системы | 20 |
| 4 | Разработка структур данных | 10 |
| 5 | Разработка алгоритмов решения частных задач | 20 |
| 3 (ТП, РП) | 86 | 6 | Реализация структур хранения данных | 11 |
| 7 | Реализация алгоритмов решения частных задач | 15 |
| 8 | Разработка пользовательского интерфейса | 5 |
| 9 | Реализация пользовательского интерфейса | 10 |
| 10 | Отладка всего комплекса | 10 |
| 11 | Исправление ошибок и недочетов | 10 |
| 12 | Разработка документации к системе | 20 |
| 13 | Итоговое тестирование системы | 5 |
| 4 (В) | 14 | 14 | Установка и настройка ПП | 14 |
| Всего | 174  чел-дн |  |  | 174  Чел-дн |

Табл.5.3 Трудоёмкости по стадиям разработки проекта.

## 

## Расчет количества исполнителей

Средняя численность исполнителей при реализации проекта разработки и внедрения ПО определяется соотношением:, где:

Qp - затраты труда на выполнение проекта (разработка и внедрение ПО),

F - фонд рабочего времени.

Величина фонда рабочего времени определяется соотношением:

, где

Т - время выполнения проекта в месяцах. T = 5 мес.;

FM - фонд времени в текущем месяце, который рассчитывается из учета общества числа дней в году, числа выходных и праздничных дней:

, где

tp - продолжительность рабочего дня;

DK - общее число дней в году;

DB - число выходных дней в году;

DП - число праздничных дней в году.





 - число исполнителей проекта.

## Календарный план-график разработки ПП

Планирование и контроль хода выполнения разработки проводится по календарному графику выполнения работ.

| Стадия разработки | Трудоемкость | Должность исполнителя | Распределение трудоемкости | Численность |
| --- | --- | --- | --- | --- |
| 1.Разработка технического задания | 24 | Ведущий инженер  Программист | 16 (65%)  8 | 1  1 |
| 2.Разработка эскизного проекта | 50 | Ведущий инженер  Программист | 30 (60%)  20 | 1  1 |
| 3.Разработка технического проекта | 86 | Ведущий инженер  Программист | 43  43 | 1  1 |
| 5.Внедрение | 14 | Ведущий инженер  Программист | 6  8 | 1  1 |
| Итого: | 174 |  |  | 2 |

Табл.5.4 Планирование процесса разработки.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Этапы |  | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 (ТЗ) | 16 | | |  | | | | | | | | | | | | | | | | | | | | | |
| 8 | |  | | | | | | | | | | | | | | | | | | | | | | |
| 2 (ЭП) |  | | | 30 | | | | |  | | | | | | | | | | | | | | | | |
|  | | | 20 | | |  | | | | | | | | | | | | | | | | | | |
| 3 (ТП) |  | | | | | | | | 43 | | | | | |  | | | | | | | | | | |
|  | | | | | | | | 43 | | | | | |  | | | | | | | | | | |
| 5 (В) |  | | | | | | | | | | | | | | 6 | |  | | | | | | | | |
|  | | | | | | | | | | | | | | 8 | | |  | | | | | | | |
| Время  (Дни) | 0 | 10 | | | 20 | 30 | | 40 | | 50 | 60 | 70 | 80 | | | 90 | | | 100 | 110 | 120 | 130 | 140 | 150 | 160 |
|  | Ведущий инженер | | | | | | | | | | | | |  | | | | | | | | | | | |
| Программисты | | | | | | | | | | | | |  | | | | | | | | | | | |

Табл.5.5 Календарный ленточный график работ.

**Вывод:** при распараллеливании работы ведущего инженера и программиста можно добиться сокращения срока разработки и внедрения программного продукта с 174 дней до 97 дней, т. е. в 1,8 раза по сравнению со временем разработки одним человеком, что близко к теоретическому значению.

## Расчёт стоимости программного продукта

Затраты на выполнение проекта состоят из затрат на заработную плату исполнителям, затрат на закупку или аренду оборудования, затрат на организацию рабочих мест, и затрат на накладные расходы.

Ниже приведены затраты на заработную плату и отчисления на социальное страхование в пенсионный фонд, фонд занятости и фонд обязательного медицинского страхования (26%). Для всех исполнителей предполагается оклад в размере 15000 рублей в месяц.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Февраль | | | Март | | |
| Исполнитель | рабочих дней | зарплата | ЕСН | рабочих дней | зарплата | ЕСН |
| 1 | 21 | 15000 | 3900 | 22 | 15000 | 3900 |
| 2 | 8 | 5714 | 1485 | 11 | 7500 | 1950 |
| Итого: | 26099 | | | 28350 | | |

Табл.5.6 Затраты на зарплату и отчисления на социальное страхование (начало).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Апрель | | | Май | | |
| Исполнитель | рабочих дней | зарплата | ЕСН | рабочих дней | зарплата | ЕСН |
| 1 | 22 | 15000 | 3900 | 20 | 15000 | 3900 |
| 2 | 20 | 13636 | 3545 | 20 | 15000 | 3900 |
| Итого: | 36081 | | | 27800 | | |

Табл.5.7 Затраты на зарплату и отчисления на социальное страхование (продолжение).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Июнь | | | ∑, руб. |
| Исполнитель | рабочих дней | зарплата | ЕСН |  |
| 1 | 10 | 7500 | 1950 |  |
| 2 | 20 | 15000 | 3900 |  |
| Итого: | 28350 | | | 146680 |

Табл.5.8 Затраты на зарплату и отчисления на социальное страхование (окончание).

Расходы на материалы, необходимые для разработки программной продукции, указаны в таблице 5.9.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование материала | Единица измерения | Кол-во | Цена за единицу, руб. | Сумма, руб. |
| 1 | Бумага А4 | Пачка 500 л. | 1 | 300 | 300 |
| 2 | Тонер для картриджа лазерного принтера Panasonic P7100 KX | шт. | 1 | 850 | 850 |
| Всего | | | | | 1150 |

Табл.5.9 Затраты на материалы.

В работе над проектом используется специальное оборудование – персональные электронно-вычислительные машины (ПЭВМ) в количестве 2 шт. Стоимость одной ПЭВМ составляет 25000 рублей. Месячная норма амортизации K = 2,7%.



Тогда за 5 месяцев работы расходы на амортизацию составят 6750 рублей.

R = 25000 ⋅ 0,027 ⋅ 2 ⋅ 5 = 6750 рублей.

Общие затраты на разработку программного продукта (ПП) составят 154580 рублей.

## Расчет экономической эффективности

Основными показателями экономической эффективности является чистый дисконтированный доход (ЧДД) и срок окупаемости вложенных средств.

Чистый дисконтированный доход определяется по формуле:

,

где T – горизонт расчета по месяцам;

t – период расчета;

Rt – результат, достигнутый на t шаге (стоимость);

Зt – затраты;

E – приемлемая для инвестора норма прибыли на вложенный капитал.

Коэффициент E установим равным ставке рефинансирования ЦБ РФ – 12% годовых (или 1% в месяц). В результате анализа рынка программной продукции, аналогичной разрабатываемой, планируется единовременная продажа ПП по окончании выполнения проекта. Планируемая цена ПП составляет рублей.

Коэффициент дисконтирования равен 1/(1 + Е) = 0.99.

В таблице 5.10. приведен расчет ЧДД по месяцам работы над проектом.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Месяц | Текущие затраты, руб. | Затраты с начала года, руб. | Текущий доход, руб. | ЧДД, руб. |
| Февраль | 27149 | 27149 | 0 | -26880,2 |
| Март | 28350 | 55499 | 0 | -54671,6 |
| Апрель | 36081 | 91580 | 0 | -89691,5 |
| Май | 27800 | 119380 | 0 | -116407 |
| Июнь | 28350 | 147730 | 183062 | 30796,45 |

Табл.5.10 Расчёт ЧДД.

Из таблицы видно, что срок окупаемости проекта составит 5 месяцев.

Рисунок. 5.1. График изменения чистого дисконтированного дохода

## Выводы

Можно прогнозировать, что проект окажется рентабельным и окупится через 5 месяцев после начала работ. Разработанный программный комплекс не имеет аналогов. В силу этого, представляется довольно сложным оценить цену программного продукта при выдвижении его на рынок. В данном расчёте стоимость продукта рассчитывается исходя из нормы рентабельности в 25%.

# Промышленная экология и безопасность

Основой проектирования безопасной техники и технологии, а также разработки комплекса мероприятий, обеспечивающего безопасные условия труда на производстве, является учет требований законодательных актов и нормативно-технической документации по охране труда.

## Анализ вредных и опасных факторов

Пользователь персональной электронно-вычислительной машины (далее ПЭВМ) имеет дело с рядом вредных и опасных факторов:

* переменные электрические и магнитные поля;
* статическое электричество;
* негативное воздействие некомфортного микроклимата;
* опасность поражения электрическим током;
* опасность возникновения пожара;
* негативное воздействие шума;
* негативное воздействие недостатка освещения;
* утомление и травматизм кистей рук.

При длительном и регулярном воздействии на организм человека, при превышении установленных норм на воздействие эти факторы могут отрицательным образом сказаться на здоровье человека.

Воздействие вышеперечисленных вредных и опасных факторов на организм человека нормируется по СанПиН 2.2.2/2.4.1340-03 ("Гигиенические требования к персональным электронно-вычислительным машинам и организации работы"). Далее перечислены основные вредные и опасные факторы и описаны способы снижения их влияния на организм человека.

### Освещенность

Естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток и обеспечивать коэффициент естественной освещенности (КЕО) не ниже 1,5% для г. Москва.

Искусственное освещение должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях в случае преимущественной работы с документами допускается применение искусственного освещения (и вместе с общим освещением устанавливаются светильники местного освещения).

Освещенность на поверхности рабочего стола должна быть 300-500 лк. Разрешается установка светильников местного освещения для работы с документами, но они не должны создавать блики на поверхности экрана и увеличивать освещенность более 500 лк.

Путем правильного расположения рабочих мест относительно источников освещения должна ограничиваться блесткость от источников освещения.

|  |  |
| --- | --- |
| Параметр | Значение |
| Яркость светящихся поверхностей, попадающих в поле зрения | < 200 кд/кв.м |
| Яркость бликов на экране ВДТ и ПЭВМ | < 40 кд/кв.м |
| Яркость потолка при применении системы отраженного освещения (Для уменьшения используются матовые неотражающие материалы и защитные колпаки.) | < 200 кд/кв.м |
| Источники света при искусственном освещении | Должны применяться преимущественно люминесцентные лампы типа ЛБ с коэффициентом пульсации не больше 5%. |

Табл.6.1 Яркость некоторых объектов.

### Электрические и магнитные поля

Практические все составные части ПЭВМ (системный блок, монитор, дисковые накопители, принтер, сканер) являются источниками электромагнитных полей, которые взаимодействуя формируют сложную электромагнитную обстановку на рабочем месте оператора.

|  |  |
| --- | --- |
| Источник | Диапазон частот  (первая гармоника) |
| **Помехогасящие фильтры в** блоках питания | 50 Гц |
| Преобразователь напряжения в импульсном блоке питания | 20 кГц – 100 кГц |
| Блок кадровой развертки и синхронизации | 48 Гц – 160 Гц |
| Блок строчной развертки и синхронизации | 15 кГц – 110 кГц |
| **Процессор** | 50 Гц – 1000 МГц |
| **Другие устройства** | 0 Гц, 50 Гц |
| **Источники бесперебойного питания** | 50 Гц, 20 кГц – 100 кГц |

Табл.6.2 Частоты электромагнитных полей.

.3 показывает примерные значения величин полей для среднестатистических рабочих мест оператора ПЭВМ.

|  |  |  |
| --- | --- | --- |
| **Наименование измеряемых параметров** | **Диапазон частот**  **5 Гц - 2 кГц** | **Диапазон частот**  **2 - 400 кГц** |
| Напряженность переменного электрического поля, (В/м) | 1,0 - 35,0 | 0,1 - 1,1 |
| Индукция переменного магнитного поля, (нТл) | 6,0 - 770,0 | 1,0 - 32,0 |

Табл.6.3 Значения величин полей для рабочих мест оператора ПЭВМ.

В нижеприводимой таблице показаны данные Шведского Института защиты от излучений об излучениях мониторов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Среднее значение** | | | **Максимальное значение** | | |
| **Расстояние** | **0,5 м** | | **0,3 м** | **0,5 м** | | **0,3 м** |
| **Направление излучения** | **по оси** | **вокруг** | **по оси** | **вокруг** | **по оси** | **по оси** |
| магнитное поле, 5Гц- 2кГц, нТл | < 200 | < 200 | < 200 | 260 | 500 | 730 |
| магнитное поле, 2- 400 кГц, нТл | < 10 | 13 | нет данных | 52 | 52 | нет данных |
| электрическое поле, 5Гц- 2кГц, В/м | < 10 | нет данных | 17 | 74 | нет данных | 152 |
| электрическое поле, 2- 400 кГц, В/м | 1,7 | 1,9 | 4,2 | 12 | 12 | 32 |

Табл.6.4 Значения величин полей для ЭЛТ мониторов.

Согласно международному стандарту TCO’03, для ЖК мониторов регламентированы следующие требования по электрическому и магнитному излучениям:

|  |  |  |
| --- | --- | --- |
| **Расстояние** | **0,5 м** | **0,3 м** |
| **Направление излучения** | **по оси** | **по оси** |
| магнитное поле, 5Гц- 2кГц, нТл | <= 200 | <= 200 |
| магнитное поле, 2- 400 кГц, нТл | <= 25 | нет данных |
| электрическое поле, 5Гц- 2кГц, В/м | <= 10 | <= 10 |
| электрическое поле, 2- 400 кГц, В/м | <= 1 | <= 1 |

Табл.6.5 Значения величин полей согласно TCO`03 для ЖК мониторов.

В 6. приводятся предельно допустимые уровни электрического и магнитного полей в соответствии с СанПиН 2.2.2/2.4.1340-03.

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид поля** | **Диапазон частот** | **Единица измерения** | **ПДУ** |
| магнитное поле | 5 Гц- 2 кГц | нТл | 250 |
| 2 кГц - 400 кГц | нТл | 25 |
| электрическое поле | 5 Гц - 2 кГц | В/м | 25 |
| 2 кГц - 400 кГц | В/м | 2,5 |

Табл.6.5 Предельно допустимые уровни электрического и магнитного полей согласно СанПиН 2.2.2/2.4.1340-03.

Проектом предусматривается использование только тех мониторов, которые удовлетворяют требованиям СанПиН 2.2.2/2.4.1340-03.

При использовании мониторов, не соответствующих высоким требованиям СанПиН 2.2.2/2.4.1340-03, необходимо использовать специальные средства защиты от электромагнитного излучения – защитные экраны.

При работе над проектом предусмотрено использование монитора Dell 2408WPF, соответствующего требованиям стандарта TCO’03, а значит удовлетворяющего стандартам СанПиНа.

### Статическое электричество

При работе монитора на экране кинескопа накапливается электростатический заряд, создающий электростатическое поле (ЭСтП). Измерения значения ЭСтП колеблются от 8 до 75 кВ/м. Разброс электростатических потенциалов пользователей колеблется в диапазоне от -3 до +5 кВ. При воздействии ЭСтП возникнают неприятные субъективные ощущения.

Помимо монитора вклад в общее электростатическое поле вносят электризующиеся от трения поверхности клавиатуры и мыши. Например, после работы с клавиатурой, электростатическое поле быстро возрастает с 2 до 12 кВ/м. На отдельных рабочих местах в области рук регистрировались напряженности статических электрических полей более 20 кВ/м.

Среднестатистическое значение электростатического потенциала для современных мониторов составляет величину 200 В, в то время как в соответствии с СанПиН 2.2.2/2.4.1340-03 допустимым является значение 500 В. В использовавшемся при работе над проектом мониторе имеется специальное антистатическое покрытие панели, которое предотвращает накопление статических зарядов.

### Электробезопасность

Возможные поражения электрическим током являются наиболее опасными факторами при работе с электрооборудованием. Напряжение в сети составляет 220В, а частота переменного тока 50Гц, и согласно ПУЭ это помещение с повышенной опасностью. Для обеспечения безопасности используются следующие средства защиты:

* заземление выносного типа, заземлению подлежат корпуса компьютеров, мониторов и периферийных устройств, причем общее сопротивление всей системы заземления должно быть меньше 4 Ом;
* изоляция проводников, сопротивление которой, согласно ПУЭ должно составлять не менее 0,5 МОм;
* УЗО (устройство защитного отключения).

Согласно ГОСТ 12.1.030-81, Защитное заземление следует выполнять преднамеренным электрическим соединением металлических частей электроустановок с "землей" или ее эквивалентом. Защитное заземление или зануление электроустановок следует выполнять: при номинальном напряжении 380 В и выше переменного тока и 440 В и выше постоянного тока - во всех случаях; при номинальном напряжении от 42 В до 380 В переменного тока и от 110 В до 440 В постоянного тока при работах в условиях с повышенной опасностью и особо опасных.

Электрическая сеть производственного помещения снабжена системой заземления. Т.к. в блоках питания и кабелях электропитания используемой компьютерной техники присутствует заземляющий провод, это обеспечивает выполнение требований по заземлению электронной техники.

Энергосеть здания оборудована цифровыми электросчётчиками, предусматривающими автоматическое отключение электропитания помещения при превышении допустимой нагрузки (а также в результате КЗ).

### Опасность возникновения пожара

Согласно существующей классификации НПБ-105-03 производственное помещение, в котором предполагается использовать систему, по пожарной опасности относится к категории «Д». Пожарная безопасность может быть обеспечена мерами пожарной профилактики и активной пожарной защитой. Под активной пожарной защитой понимаются меры, обеспечивающие успешную борьбу с возникающими пожарами. На случай пожара необходимо предусмотреть безопасную эвакуацию людей.

Для тушения пожаров в производственном помещении необходимо применять углекислотные и порошковые огнетушители, которые обладают высокой эффективностью борьбы с огнем и возможностью тушения электроустановок.

Исходя из норм пожарной безопасности, в здании с ПЭВМ расположены внутренние средства пожаротушения, такие как пожарные краны, средства первичного пожаротушения. Также в помещении установлена пожарная сигнализация, которая позволяет оповестить дежурный персонал о пожаре. В качестве пожарных сигнальных датчиков в машинном зале устанавливаются дымовые фотоэлектрические извещатели.

### Шум

Шум создает значительные нагрузки на нервную систему человека, оказывает на него психологическое воздействие и снижает производительность труда. Источниками шума в помещении являются механические устройства и внутренние вентиляторы ЭВМ, а также шум от общеобменной вентиляционной установки.

Продолжительное действие сильного шума вызывает общее утомление, может привести к ухудшению слуха, а иногда и к глухоте, нарушается процесс пищеварения, происходит изменение объема внутренних органов. Значения допустимых уровней шума согласно СанПиН 2.2.2/2.4.1340-03 приведены в 7.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Уровни звукового давления в дБ в октавных полосах со среднегеометрическими частотами, Гц | | | | | | | |
| 63 Гц | 125 Гц | 250 Гц | 500 Гц | 1000 Гц | 2000 Гц | 4000 Гц | 8000 Гц |
| 71 дБ | 61 дБ | 54 дБ | 49 дБ | 45 дБ | 42 дБ | 40 дБ | 38 дБ |

Табл.6.7 Значения допустимых уровней шума согласно СанПиН 2.2.2/2.4.1340-03.

Измерение уровня звука и уровней звукового давления проводится на расстоянии 50 см от поверхности оборудования и на высоте расположения источника(ков) звука.

В данном случае (разработка программного обеспечения) источниками наибольшего шума могут являться матричные принтеры, однако, они применяются сейчас очень редко (в основном, в виде специализированных устройств), а также внешние источники шума - шум с улицы, из соседних помещений. Т.о. постоянные источники шума, превышающего нормы, отсутствуют.

При постоянной работе на ПЭВМ уровень шума на рабочем месте не должен превышать 50дБА. В случае наличия шумов, превышающих нормы, принимаются меры по их устранению. Такими мерами, как правило, являются применение стеклопакетов для звукоизоляции от внешнего шума, а также изменение шумовых характеристик самой ЭВМ путём замены системы охлаждения на безвентиляторную, либо применением низкооборотных вентиляторов.

Также используются звукопоглощающие материалы с максимальными коэффициентами звукопоглощения в области частот 63 - 8000 Гц для отделки помещений, подтвержденных специальными акустическими расчетами.

### Вибрация

Уровень вибрации на рабочем месте не должен превышать допустимых норм вибрации.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Среднегеометрические частоты  Октавных полос, Гц | Допустимые значения по осям X и Y | | | |
| по виброускорению | | по виброскорости | |
| м/с2 | дБ | м/с | дБ |
| 2 | 53 | 25 | 45 | 79 |
| 4 | 53 | 25 | 22 | 73 |
| 8 | 53 | 25 | 11 | 67 |
| 16 | 10 | 31 | 11 | 67 |
| 31,5 | 21 | 37 | 11 | 67 |
| 63 | 42 | 43 | 11 | 67 |
| Корректированные значения  и их уровни | 93 | 30 | 20 | 72 |

Табл.6.8 Допустимые нормы вибрации на рабочих местах с ВДТ и ПЭВМ.

При превышении указанных норм следует принять меры по уменьшению вибрации. Такими мерами могут быть:

* применение демпфирующих материалов в качестве прокладок на пути распространения вибрации, например, между полом и рабочим столом, ножки системного блока компьютера и т.д.;
* уход от резонансных режимов (в том случае, если частота возмущающего воздействия и частота собственных колебаний системы сопоставимы);

### Травматизм

В качестве основного устройства ввода для ПЭВМ используется клавиатура. Длительная работа на клавиатуре может вызвать значительное утомление пальцев и кистей рук оператора. В соответствии с требованиями СанПиН 2.2.2/2.4.1340-03 клавиатура должна удовлетворять следующим требованиям:

* исполнение в виде отдельного устройства с возможностью свободного перемещения;
* опорное приспособление, позволяющее изменять угол наклона поверхности клавиатуры в пределах 5-15 градусов;
* высота среднего ряда клавиш не более 30 мм;
* расположение часто используемых клавиш в центре, внизу и справа, редко используемых - вверху и слева;
* выделение цветом, размером, формой и местом расположения функциональных групп клавиш;
* минимальный размер клавиш 13 мм, оптимальный - 15 мм;
* клавиши с углублением в центре и шагом 19 плюс - минус 1 мм;
* расстояние между клавишами не менее 3 мм;
* одинаковый ход для всех клавиш с минимальным сопротивлением нажатию 0,25Н и максимальным - не более 1,5Н;
* звуковую обратную связь от включения клавиш с регулировкой уровня звукового сигнала и возможности ее отключения.

При создании данного проекта использовалась клавиатура BTC 8193, удовлетворяющая требованиям СанПиН 2.2.2/2.4.1340-03.

### Микроклимат

В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

В соответствии с установленными выше данными, можно определить оптимальные нормы микроклимата для рабочего помещения программиста (разработчика) и рабочего места пользователя программы.

Результирующие данные по оптимальным нормам микроклимата для разработчика и пользователя программного продукта приводятся в

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Работник | Оптимальные параметры | | |
| Температура воздуха, °C | Относительная влажность воздуха, % | Скорость движения воздуха, м/с |
| Холодный | Разработчик | 22-24 | 40-60 | 0,1 |
| Пользователь | 21-23 | 40-60 | 0,1 |
| Теплый | Разработчик | 23-25 | 40-60 | 0,1 |
| Пользователь | 22-24 | 40-60 | 0,2 |

Табл.6.9 Оптимальные нормы микроклимата для разработчика и пользователя программного продукта.

Если параметры микроклимата не соответствуют установленным нормам, то необходимо применять системы кондиционирования или вентиляции для приведения их в норму. При этом, поскольку работа за компьютером требует высокого сосредоточения и концентрации, более предпочтительными являются приточно-вытяжные системы кондиционирования, автоматически поддерживающие требуемый режим.

Еще одним вредным фактором при работе с ЭВМ является запыленность помещения. В любом рабочем помещении есть частицы пыли. Однако персональные компьютеры за счет электризации и накопления статического заряда еще и притягивают поток этих частиц. Избежать запыленности позволяет применение общеобменной системы вентиляции.

Уровни положительных и отрицательных ионов в воздухе должны соответствовать нормам, приведенным в

|  |  |  |
| --- | --- | --- |
| Уровни | Число ионов в 1 см куб. воздуха | |
| n+ | n- |
| Минимально необходимые | 400 | 600 |
| Оптимальные | 1500-3000 | 3000-5000 |
| Предельно допустимые | 50000 | 50000 |

Табл.6.10 Уровни ионизации воздуха помещений при работе на ВДТ и ПЭВМ.

Содержание вредных химических веществ в помещениях с ПЭВМ не должно превышать “ПДК загрязняющих веществ в атмосферном воздухе населенных мест” (СанПиН 2.1.6.575-96).

### Питьевая вода

Производственное помещение подключено кцентрализованной системе питьевого водоснабжения.В соответствии сСанПиН 2.1.4.1074-01 «Питьевая вода. Гигиенические требования к качеству воды централизованных систем питьевого водоснабжения. Контроль качества», очистка, фильтрация и иная подготовка воды производится непосредственно перед подачей её в систему центрального водоснабжения с тем, чтобы в результате доставки к производственному помещению она не потеряла своих качеств.

Для дополнительной очистки воды возможно использование фильтров. Например, питьевая вода, очищенная фильтром Аквафор «Кувшин» соответствует СанПиН 2.1.4.1074-01, что гарантирует пригодность её к употреблению.

## Расчет освещения

В системе освещения рабочего помещения предусмотрено использование как естественного света, поступающего через оконные проёмы, так и искусственного, создаваемого лампами общего освещения и локальными светильниками на рабочих местах.

### Расчет площади светопроемов

Вычисляем нормированное значение КЕО. Для заданного II разряда работ принимаем , а для центрального региона . Таким образом, 

Определяем суммарную площадь светопроемов для заданной нормированной освещенности при боковом одностороннем освещении по формуле:

,

где SП – площадь пола помещения, м2;

;

eN – нормированное значение КЕО,

η0 – световая характеристика окна, определяется по таблицам СНиП на основании

отношений LП/В и В/h1:

η0=10;

К3 – коэффициент запаса, учитывающий загрязнение светопропускающего материала светового проема, зависит от типа помещения и от расположения стекол. При вертикальном расположении К3=1,2;

К3Д – коэффициент, учитывающий затемнение окон противостоящими зданиями. При отсутствии противостоящих зданий К3Д=1;

r1 – коэффициент, учитывающий отраженный свет. Принимаем r1=1,2;

τ0 – общий коэффициент светопропускания светового проема.



τ1 – коэффициент светопропускания материала. Для стеклопакета 0,8;

τ2 – коэффициент, учитывающий потери света в переплетах окна. Для стекложелезобетонных панелей с пустотелыми стеклянными блоками 0,8.

τ3 – коэффициент, учитывающий потери света в несущих конструкциях. При отсутствии несущих конструкций 1.

τ4 – коэффициент, учитывающий потери света в солнцезащитных устройствах. При отсутствии таковых 1.



Вычисляем необходимую суммарную площадь световых проемов:



Определяем площадь одного светового проема, имеющегося в рабочем помещении.

м2

Тогда количество световых проемов, необходимых при данном значении нормированного КЕО, расчитывается по формуле:



Таким образом, рассматриваемое помещение удовлетворяет требованиям к минимальному значению КЕО.

На рисунке 6.1 приводится план помещения с учетом необходимых светопроемов.

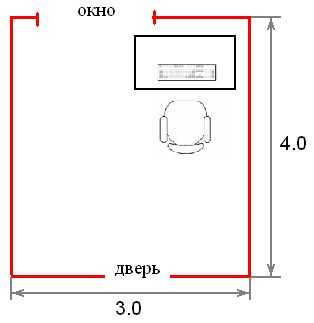


Рисунок 6.1. План помещения с учетом необходимых светопроемов.

В зависимости от погодных условий и времени года, дневное естественное освещение на улице варьируется от 5000 лк до 100000 лк. Таким образом, освещённость рабочего места только лишь от естественного бокового освещения может составлять 90-1800 лк. При превышении предельного значения освещённости (500 лк) необходимо задействовать средства снижения уровня естественного освещения – жалюзи, шторы. На случай недостатка освещённости рабочего места предусмотрено искусственное освещение – общее (люстры, лампы дневного освещения, в т.ч. вмонтированные в потолок) и местное (локальные, индивидуальные светильники).

### Расчет искусственного освещения

#### Общее освещение

Необходимо решить следующие вопросы:

1. определить типы ламп и светильников;
2. выбрать расположение светильников и определить потребность в их количестве

Для освещения рабочего помещения с ПЭВМ используются люминесцентные лампы, обычно типа ЛБ 20/40/80 или их зарубежные аналоги. Для освещения проектируемого помещения предусмотрено использование ламп ЛБ 40. Расположение светильников – сплошными рядами.

Для расчета искусственного освещения системы общего освещения используется метод светового потока. Световой поток определяется по формуле

, где

 – световой поток лампы в люменах;

 – нормируемая освещенность, лк;

 – площадь помещения, м2;

k – коэффициент запаса;

z – отношение средней освещенности к минимальной, принимается z = 1,1÷1,15;

N – число светильников;

η– коэффициент использования светового потока ламп.







Коэффициент использования светового потока зависит от типа светильника, коэффициентов отражения потолка ρП, стен ρС и индекса помещения (i), учитывающего геометрические параметры помещения и высоту подвеса светильника:



, где

 – высота подвеса светильника над рабочей поверхностью;

 – высота помещения;

 – высота подвеса светильника от потолка;

 – высота рабочего места.









ρП = 70%

ρС = 50%

η = 0,51





Таким образом, необходимо установить 4 лампы общего освещения (при этом световой поток будет меньше расчетного на 3%). Предполагается организовать 2 светильника с 2 лампами в каждом. Светильники будут расположены на некотором расстоянии друг от друга. Предусматривается раздельное управление светильниками, чтобы их можно было включать по отдельности. Длина одного светильника – 1 м, ширина – 0.4 м.

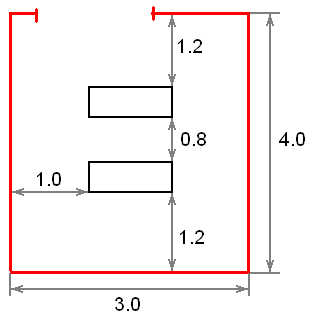


Рисунок 6.2. Расположение светильников общей искусственной системы освещения.

#### Местное освещение

Согласно СНиП 23-05-95 для местного освещения (в составе комбинированного освещения) следует использовать светильники с непросвечивающимми отражателями. Светильники местного освещения следует располагать так, чтобы их светящие элементы не попадали прямо в поле зрения работников как данного так и других рабочих мест. Выражение для освещенности данной точки “a” поверхности наблюдаемого объекта определяется выражением:

.

Здесь Jсв – сила света, излучаемого светильником, кд/м2; lса – длина пути светового луча от светящего элемента до точки “a” наблюдаемого объекта, м; - угол, образуемый световым лучом в направлении от светящего элемента к точке “a” и нормалью к наблюдаемой поверхности в точке “a”, рад. Расположение светильника на рабочем месте в общем случае показано на рис. 6.3.



Рисунок 6.3. Схема расположения светильника на рабочем месте.

Светильник имеет следующие параметры:

Коэффициент отражения отражающей поверхности отражателя светильника 

Коэффициент, определяющий отношение отражающей поверхности рефлектора светильника к его полной поверхности 

Диаметр светильника 

Высота расположения центра светящейся поверхности лампы относительно нижнего среза светильника 

В точке “a” светильник местного освещения должен создавать освещенность, равную нормативному значению для местного освещения. В случае применения местного освещения в составе системы комбинированного освещения для работы с ЭВМ уровень освещённости рабочего места должен составлять 1000лк с отклонением в пределах (–10%) - (+20%). Таким образом, уровень освещённости, создаваемый светильником должен быть равен 700лм.

Определим силу света, требуемую от светильника местного освещения:





Для данного значения светового потока можно выбрать лампу накаливания мощностью 80 Вт. Для изменения уровня освещенности рабочего места от лампы местного освещения возможно изменить ориентацию светильника. Предусматривается использование светильников с возможностью регулирования яркости, либо отдельных устройств для обеспечения такого регулирования.

Суммарный уровень освещённости рабочего места равен сумме отдельных составляющих: уровень освещённости от естественного освещения, искусственного общего и местного освещений.



Спроектированная система освещения позволит всегда получать на рабочем месте уровень освещенности в пределах нормативного.

# Заключение

В результате выполнения квалификационной работы был модифицирован и реализован алгоритм декомпозиции конечных вероятностных автоматов.

Для демонстрации работы данного алгоритма было разработано Windows-приложение, позволяющее:

* инициализировать исходный вероятностный автомат;
* производить декомпозицию заданного автомата;
* моделировать работу, как самого автомата, так и сети, полученной в результате декомпозиции;
* производить импорт и экспорт вероятностного автомата и результирующей сети в формате XML.

Разработанный алгоритм и основные сущности, связанные с предметной областью (вероятностный автомат, сеть и т.п.) были реализованы в виде .NET-библиотеки, что позволяет использовать её в составе программных комплексов, предназначенных для анализа дискретных систем.

Также, в ходе исследований, проведённых в рамках данной квалификационной работы, были изучены количественные и качественные характеристики разработанного алгоритма.

# Перспективы развития проекта

В качестве дальнейшего улучшения и развития проекта можно рассмотреть следующие идеи:

* расширить набор критериев для отбора множества ортогональных разбиений в ходе декомпозиции автомата, рассмотреть возможность объединения данных критериев в экспертную систему;
* рассмотреть варианты оптимизации разработанного алгоритма, с целью улучшения временных характеристик работы сети, полученной в результате декомпозиции;
* логическим продолжением данной квалификационной работы является разработка комплекса анализа дискретных систем, использующего разработанную библиотеку в качестве основного инструмента исследования.

# Список литературы.

1. Баранов С.И. Синтез микропрограммных автоматов. - 2-е изд., перераб. и доп. Л. Энергия Ленингр. отд-ние, 1979.

2. Лазарев В. Г., Пийль Е. И. Синтез управляющих автоматов. — 3-е изд., переработанное. И дополненное. Москва, издательство Энергоатомиздат, 1989.

3. Санковский Е.А. Шаталов А.С. Теория автоматического управления. М: Высшая школа, 1977.

4. Бухараев Р. Г.. Вероятностные автоматы. Итоги науки и техн. Сер. Теор. вероятн. Мат. стат. Теор. кибернет., 1978, 79–122.