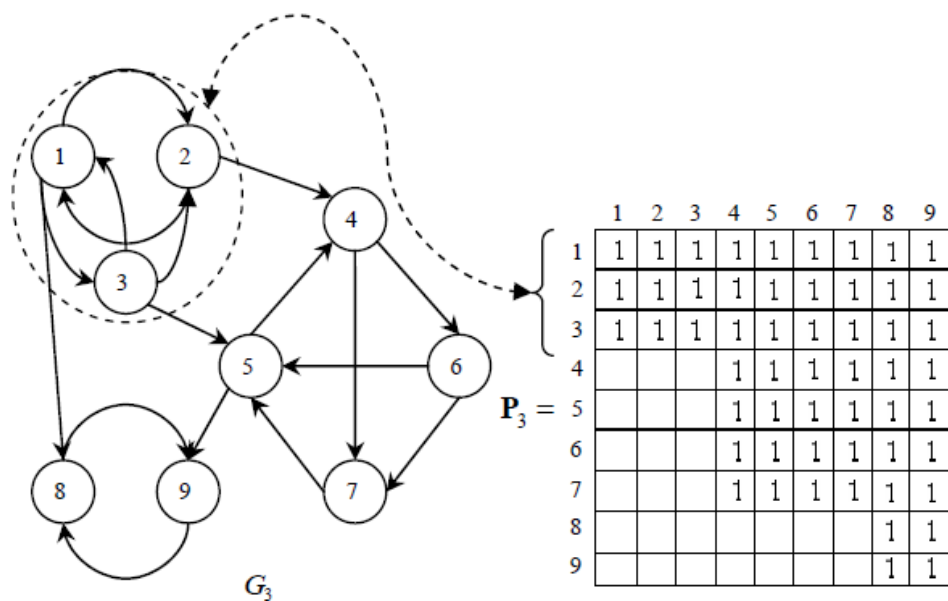


Н.Ф. Гусарова, Н.В. Добренко
ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО

Н.Ф. Гусарова, Н.В. Добренко

ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ

Учебно-методическое пособие

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 09.03.02 «Информационные системы
и технологии» в качестве учебно-методического пособия для реализации
основных профессиональных образовательных программ
высшего образования бакалавриата

 **УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2019

Гусарова Н.Ф., Добренко Н.В. **Теория систем и системный анализ.** – СПб: Университет ИТМО, 2019. – 84 с.

Пособие охватывает курс лабораторных работ, выполняемых в рамках курса «Теория систем и системный анализ». Рассматриваются вопросы, связанные с построением модели структуры информационной системы, оценкой ее структурной сложности, проектных и динамических характеристик. Особое внимание уделяется изучению особенностей информатизации деятельности организации при различных подходах к ее описанию.

Пособие предназначено для бакалавров по направлению подготовки 09.03.02 Информационные системы и технологии по программе «Мобильные и сетевые технологии» и содержит материалы лабораторных работ по дисциплине «Теория систем и системный анализ».



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2019

© Гусарова Н.Ф., Добренко Н.В., 2019

Оглавление

Введение.....	4
Лабораторная работа №1. Построение модели структуры информационной системы.....	6
Лабораторная работа №2. Оценка структурной сложности информационной системы.....	24
Лабораторная работа №3. Оценка проектных характеристик архитектурной модели информационной системы	36
Лабораторная работа №4. Анализ динамических характеристик информационной системы на основе ее функциональной модели.....	46
Лабораторная работа №5. Изучение особенностей информатизации дея- тельности организации при различных подходах к ее описанию.....	56
Лабораторная работа №6. Оценка характеристик бизнес-процесса с применением аппарата сетей Петри	64
Лабораторная работа №7. Интегральная оценка качества бизнес-процесса с применением метрик.....	74
Литература	83

Введение

Академик Н.Н. Моисеев приводит следующее определение системного анализа: «Системный анализ – это совокупность методов, основанных на использовании ЭВМ и ориентированных на исследование сложных систем – технических, экономических, экологических и т. д. Результатом системных исследований является, как правило, выбор вполне определенной альтернативы: плана развития региона, параметров конструкции и т. д. Поэтому истоки системного анализа, его методические концепции лежат в тех дисциплинах, которые занимаются проблемами принятия решений: теории операций и общей теории управления» .

Эффективность решения проблем с помощью системного анализа определяется структурой решаемых проблем. Рассмотрение категорий системного анализа создает основу для логического и последовательного подхода к проблеме принятия решений. Как показывает практика, эффективность системно-аналитического подхода особенно явно проявляется при построении и использовании информационных систем.

В предлагаемом пособии представлен цикл лабораторных работ, выполняемых в рамках курса «Теория систем и системный анализ» для основных профессиональных образовательных программ высшего образования бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии. Для работы с пособием необходимо изучение теоретического материала в рамках тем программы данной дисциплины.

Пособие содержит 2 цикла лабораторных работ, направленных на приобретение умений и навыков, а также освоение и развитие компетенций. Первый цикл (4 лабораторных работы) связан с приобретением навыков системно-аналитической оценки реальной информационной системы, а второй цикл (3 лабораторных работы) – связан с приобретением навыков системно-аналитической оценки информатизации деятельности организации.

Описание каждой лабораторной работы содержит основные теоретические сведения, задание на работу и состав отчета по работе, а также пример выполнения работы. Студенты выполняют лабораторные работы самостоятельно или в группах. Полученные при этом результаты оформляются в отчеты, подлежащие защите в течение семестра, но не позднее зачетной недели. В лабораторных работах предусматривается текущий контроль, проводимый в форме доклада по результатам проделанной работы обучающегося и ответов на вопросы преподавателя. Промежуточный контроль проводится в устной форме. В конце пособия приводится список рекомендуемых источников литературы.

Общие методические рекомендации для выполнения представленных в пособии лабораторных работ:

1. Выбрать информационную систему для построения ее модели и выполнения последующих лабораторных работ. Согласовать выбранную систему с преподавателем.

2. Описать общее назначение системы.

3. Построить модель структуры системы на уровне программно-технической архитектуры.

4. Построить модель структуры этой же системы на любом другом уровне моделирования (например, на уровне модели предметной области).

5. Описать содержательно, каким образом в построенных моделях учтены особенности построения модели структуры.

Выполнение лабораторных работ направлено на закрепление теоретических знаний и получение практических навыков при изучении данной дисциплины. В результате освоения материалов пособия обучающийся приобретает следующие умения и навыки:

- уметь сформулировать основные понятия и этапы системного анализа;
- выбирать и использовать методы системного анализа для решения проблемных ситуаций;
- оценивать проблемные ситуации, возникающие при разработке архитектур интеллектуальных систем и процессов обработки данных;
- управлять декомпозицией проблемных ситуаций при построении интеллектуальных систем;
- структурировать проблемные ситуации, возникающие при разработке архитектур интеллектуальных систем;
- владеть базовыми методами декомпозиции проблемных ситуаций;
- владеть навыками адекватного выбора методов моделирования процессов обработки данных;
- владеть навыками адекватного выбора количественных методов описания проблемных ситуаций.

Распределение трудозатрат студентов в аудитории и в процессе СРС представлено в соответствии с программой изучаемой дисциплины. В рамках самостоятельной работы студентам рекомендуется отвести на изучение теоретических материалов по плану лекций каждого раздела ориентировочно по 20 часов, на подготовку и выполнение лабораторных работ и подготовку к промежуточному контролю также ориентировочно по 40 часов, и по 23 часов на самостоятельное изучение дополнительных источников информации.

Авторы выражают благодарность студентам Ключкову А.П. и Луцеву А.Р. за помощь при подготовке материалов для данного пособия.

Лабораторная работа №1. ПОСТРОЕНИЕ МОДЕЛИ СТРУКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ

ЦЕЛЬ РАБОТЫ

Научиться анализировать архитектуру существующей информационной системы и представлять ее в виде модели структуры.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Структура системы – совокупность необходимых и достаточных для достижения цели отношений (связей) между элементами системы.

Особенности построения модели структуры:

1. Понятие элементарности можно определить по-разному: что с одной точки зрения является элементом, с другой – оказывается подсистемой, подлежащей дальнейшему разделению;
2. Модель структуры является целевой, и для различных целей один и тот же объект потребуется разбить на разные части.
3. Объекты системы, их свойства и связи составляют обменное соотношение, т.е. один и тот же факт можно рассматривать как характеристику объекта, его свойства или его связи с другим объектом. Отсюда возникают различные варианты модели структуры. Критерии выбора:
 - при отображении внешнего мира человек старается действовать по схеме "максимальная вариативность объектов – конечный набор отношений между ними"
 - желательно, чтобы в конкретной системе по большинству типов связей структурирование было одинаковым.
4. Для информационных систем характерна следующая номенклатура моделей:
 - Сама предметная область
 - Модель предметной области
 - Логическая модель данных
 - Физическая модель данных
 - Функциональная модель приложения
 - Программная (программно-техническая) архитектура приложения
 - Собственно приложение
 - другие варианты

На каждом уровне фигурирует своя номенклатура объектов и свои типы связей. При построении модели конкретного уровня важно предусматривать, каким образом будут осуществляться переходы на другие уровни моделирования.

Одним из наиболее распространенных подходов к моделированию информационной системы является использование уровня программно-технической архитектуры приложения – выделение в качестве ее элементов программных или аппаратных компонентов, реализующих основные информационные процессы: ввода, вывода, передачи, обработки и хранения информации, а в качестве связей этих элементов – потоков данных, передаваемых между элементами. Такая модель позволяет в дальнейшем легко перейти к более строгому (например, используя стандарт UML) представлению информационной системы на уровне технической (элементы – аппаратные узлы системы; связи – физические каналы связи), программной (элементы – программные компоненты: модули, классы и т.п.; связи – методы передачи данных между компонентами: сообщения, вызовы и т.п.) или информационной (элементы – структуры данных, связи – отношения между элементами данных в разных структурах) архитектуры и описать функциональные требования к элементам системы.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

1. Выбрать информационную систему для построения ее модели и выполнения последующих лабораторных работ. Согласовать выбранную систему с преподавателем. Система должна отвечать следующим требованиям:
 - a. Наличие полноценной технической документации (рекомендуется использовать системы с открытым кодом или имеющие поддержку не только на уровне пользователя, но и на уровне разработчиков).
 - b. Размер и сложность системы должны быть достаточны для нетривиального анализа. Рекомендуется выбор систем, модель состава которых включает не менее 15 элементов, для которых можно выделить хотя бы два уровня иерархии (элементы могут быть сгруппированы в подсистемы).
 - c. Система не является уникальной – существуют другие системы, решающие такие же или близкие задачи.
2. Описать общее назначение системы.
3. Построить модель структуры системы на уровне программно-технической архитектуры, включающую следующие артефакты:
 - a. Графическое представление модели структуры как ориентированного графа, вершинами которого являются элементы системы, а дугами – связи между элементами. В случае наличия подсистем они могут быть обозначены на графе как выделение нескольких вершин.

- б. Описание элементов системы в виде таблицы. Отдельный элемент должен быть выделен таким образом, чтобы он выполнял строго одну функцию, которая может быть отнесена к одному из базовых типов информационных процессов: ввод, вывод, обработка, хранение, передача информации. Для интерфейсных элементов допускается выделение элемента, одновременно реализующего и функцию ввода, и функцию вывода. Для элементов, сгруппированных в подсистемы необходимо отметить принадлежность к подсистеме.
 - с. Описание связей элементов в виде таблицы. Для связей должна быть указана характеристика передаваемых данных (формат данных, протокол, передаваемые параметры в вызове метода, управляющий сигнал и т.п.). Для формального представления всей совокупности связей системы требуется построить матрицу смежности. Напомним, что матрица смежности графа G с конечным числом вершин n – это квадратная матрица A размера n , в которой значение элемента a_{ij} равно числу рёбер из i -й вершины графа в j -ю вершину.
4. Построить модель структуры этой же системы на любом другом уровне моделирования (например на уровне модели предметной области или функциональной модели приложения). Форма представления результатов моделирования в этом случае выбирается свободно.
 5. Описать содержательно, каким образом в построенных моделях учтены особенности построения модели структуры.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Для выполнения работы выбраны графические библиотеки OpenGL и Apple Metal. Весь анализ проводился в контексте операционной системы macOS.

OpenGL (Open Graphics Library) – спецификация, определяющая платформонезависимый (независимый от языка программирования) программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику.

Metal – это низкоуровневый, низкозатратный аппаратно-ускоренный 3D-графический и вычислительный интерфейс программирования приложений (API), разработанный Apple Inc.

Программно-техническая архитектура OpenGL и Metal представлена на рис. 1.1 и 1.2 соответственно. Описание элементов, связей и матрица смежности для системы OpenGL представлены в табл. 1.1, 1.2 и 1.3 соответственно. Описание элементов, связей и матрица смежности для системы

Metal представлены в табл. 1.4, 1.5 и 1.6 соответственно. Архитектура OpenGL и Metal на уровне приложений представлена на рис. 1.3 и 1.4 соответственно.

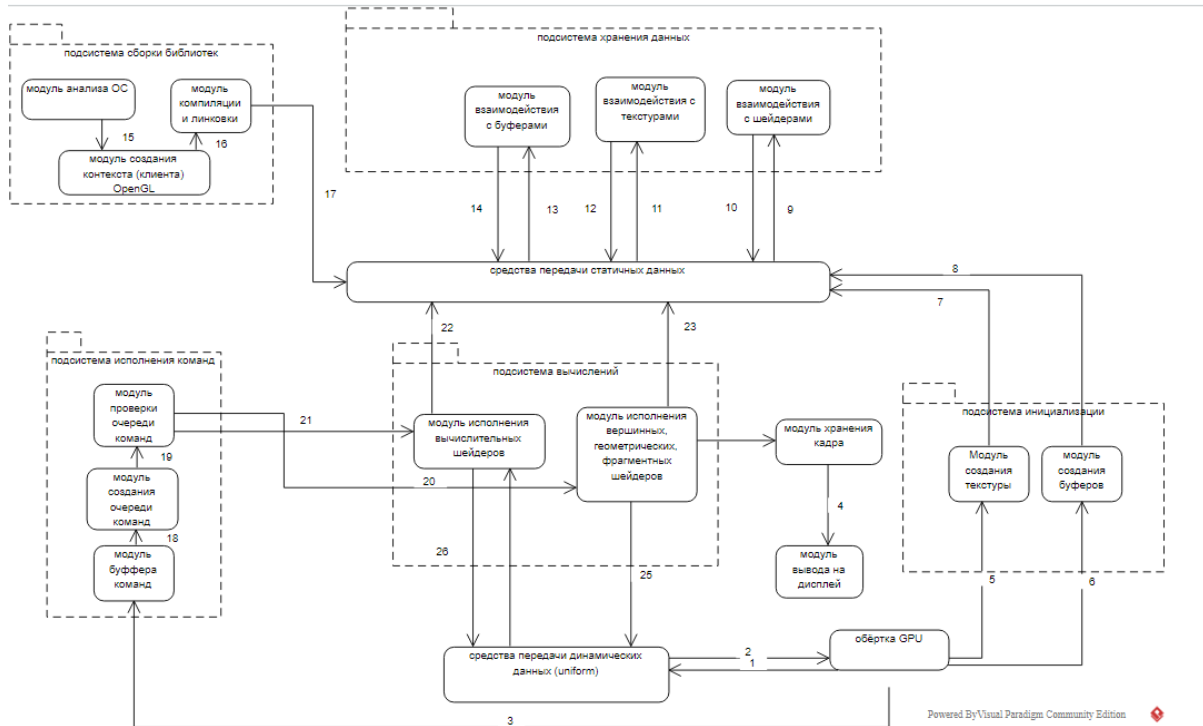


Рис. 1.1. Программно-техническая архитектура OpenGL

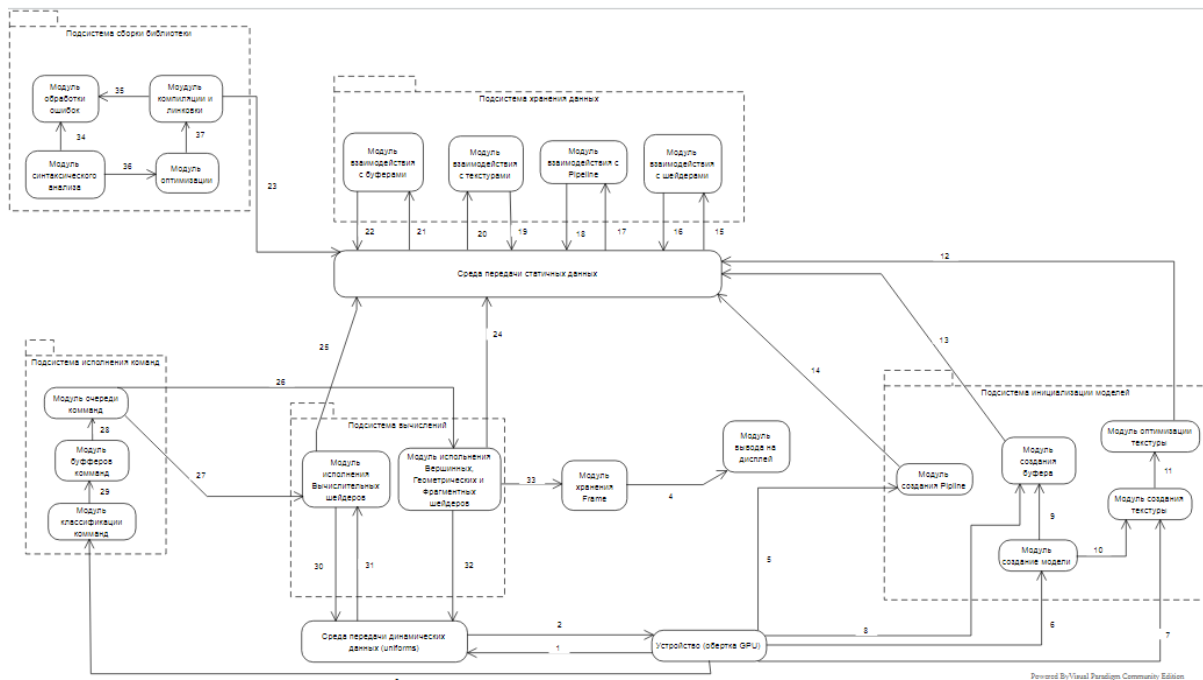


Рис. 1.2. Программно-техническая архитектура Metal

Таблица 1.1. Описание элементов системы OpenGL

Но ме р	Название эле- мента	Назначение	Тип инфор- мационного процесса	Принадлеж- ность подси- стеме
1	Модуль компиляции и линковки	Осуществляет сборку всех шейдеров в одну библиотеку, для единого доступа ко всем шейдерам	Обработка	Подсистема сборки библиотеки
2	Модуль взаимодействия с буферами	Осуществляет доступ к буферам моделей: полигонам, нормолям	Хранение	Подсистема хранения данных
3	Модуль взаимодействия с текстурами	Осуществляет доступ к текстурным моделям	Хранение	Подсистема хранения данных
4	Модуль взаимодействия с шейдерами	Осуществляет доступ к шейдерам для выполнения команд	Хранение	Подсистема хранения данных
5	Среда передачи статических данных	Осуществляет передачу данных из подсистемы хранения данных к другим модулям и обратно	Передача	
6	Модуль создания очереди команд	Осуществляет организацию правильного порядка команд (согласно переданным), для передачи в подсистему вычислений. Получать информацию может многопоточно с нескольких буферов	Передача	Подсистема исполнения команд
7	Модуль буферов команд	Осуществляет заполнение буферов команд для передачи в очередь команд (выполняется только однопоточно)	Передача	Подсистема исполнения команд
8	Модуль проверки очереди команд	осуществляет проверку созданной очереди команд	Обработка	Подсистема исполнения команд

9	Модуль исполнения Вычислительных шейдеров	Осуществляет многопоточное вычисление переданной задачи, сохраняя результат в uniforms	Обработка	Подсистема вычислений
10	Модуль исполнения Вершинных, Геометрических и Фрагментных шейдеров	Осуществляет прохождение данных, из переданных буферов и uniforms, через графический конвейер: вершинный шейдер -> геометрический -> фрагментный	Обработка	Подсистема вычислений
11	Модуль хранения Frame	Осуществляет хранение полученных результатов из модуля 14 для последующей отрисовки на дисплее	Хранение	
12	Модуль вывода на дисплей	Осуществляет вывод frame	Вывод	
13	Модуль создания буфера	Осуществляет выделение необходимого количества памяти на GPU для хранения данных	Ввод	Подсистема инициализации моделей
14	Модуль создания текстуры	Осуществляет выделение необходимого количества памяти на GPU для хранения данных	Ввод	Подсистема инициализации моделей
15	Среда передачи динамических данных (uniforms)	Осуществляет общий блок памяти для быстрой передачи изменяемых данных между CPU и GPU	Передача	
16	Устройство (обертка GPU)	Осуществляет интерфейс GPU	Ввод, Вывод	
17	модуль анализа ОС	производит анализ ОС и устанавливает наиболее подходящий формат пикселей исходя из полученной информации	Обработка	Подсистема сборки библиотеки

18	модуль создания контекста	производится создание контекста производства, который связывает OpenGL с графическими компонентами GDI	Обработка	Подсистема сборки библиотеки
----	---------------------------	--------------------------------------------------------------------------------------------------------	-----------	------------------------------

Таблица 1.2. Описание связей элементов системы OpenGL

Номер связи на схеме	Название связи	Характеристика передаваемых данных
1	Получение данных Uniforms с GPU	Получение данных из общего буфера между CPU и GPU
2	Передача данных Uniforms с CPU	Передача данных в общий буфер между CPU и GPU
3	Передача команд для выполнения текущего задания	Передача набора команд
4	Передача кадра для вывода на дисплей	Передача матрицы изображения
5	Передача текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек
6	Передача предобработанного буфера	Полигоны, нормали
7	Сохранение текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек (точек не больше, чем изначально)
8	Запись шейдера	Программный код на языке OpenGL Shading Language
9	Получение шейдера	Ссылка на функции из программного кода на языке OpenGL Shading Language
10	Получение буфера	Ссылки на структуры с полигонами, нормальями
11	Запись текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек

12	Получение текстуры	Получение ссылки на матрицу изображения и привязанных к конкретным местам текстуры точек
13	Сохранение буфера	Полигоны, нормали
14	Запись буфера	Полигоны, нормали
15	передача информации об установленном формате пикселей	данные о формате пикселей (PIXELFORMATDESCRIPTOR)
16	Передача программного кода	Программный код на языке OpenGL Shading Language
17	Сохранение шейдеров	Программный код на языке OpenGL Shading Language
18	Отправка команды в очередь	Ссылка на шейдер
19	отправка очереди команд на проверку	ссылка на очередь шейдеров
20	Отправка команды для отрисовки	Ссылка на шейдер
21	Отправка команды для вычисления	Ссылка на шейдер
22	Запрос необходимых данных для выполнения текущей команды отрисовки	Ссылки на структуры с полигонами, нормальями и текстурами
23	Запрос необходимых данных для выполнения текущей команды вычисления	Ссылки на структуры с полигонами, нормальями и текстурами
24	Отправка вычисленных uniforms	Разнородные данные из общего буфера между CPU и GPU
25	Запрос на uniforms	Разнородные данные из общего буфера между CPU и GPU
26	Запрос на uniforms	Разнородные данные из общего буфера между CPU и GPU

Таблица 1.3. Матрица смежности элементов OpenGL

	Модуль компиляции шейдеров	Модуль взаимодействия с буферами	Модуль взаимодействия с текстурными шейдерами	Модуль взаимодействия с шейдерами	Среды передачи статических данных	Модуль создания очереди команд	Модуль буферов команд	Модуль проверки очереди команд	Модуль исполнения Вычислительных шейдеров	Модуль исполнения Вершинных, Геометрических и Фрагментных шейдеров	Модуль хранения Frame	Модуль вывода на дисплей	Модуль создания буфера	Модуль создания текстуры	Среды передачи динамических данных (uniforms)	Устройство (обертка GPU)	модуль анализа ОС	модуль создания контекста
Модуль компиляции шейдеров	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Модуль взаимодействия с буферами	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Модуль взаимодействия с текстурными шейдерами	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Модуль взаимодействия с шейдерами	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Среды передачи статических данных	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Модуль создания очереди команд	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Модуль буферов команд	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Модуль проверки очереди команд	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Модуль исполнения Вычислительных шейдеров	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Модуль исполнения Вершинных, Геометрических и Фрагментных шейдеров	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Модуль хранения Frame	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Модуль вывода на дисплей	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Модуль создания буфера	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Модуль создания текстуры	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Среды передачи динамических данных (uniforms)	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
Устройство (обертка GPU)	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0
модуль анализа ОС	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
модуль создания контекста	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Таблица 1.4. Описание элементов системы Metal

Но ме р	Название эле- мента	Назначение	Тип информа- ционного про- цесса	Принадлеж- ность подси- стеме
1	Модуль обра- ботки ошибок	Осуществляет генерацию сообщений об ошибке по вызванному исключению	Вывод	Подсистема сборки библиотеки
2	Модуль компи- ляции и лин- ковки	Осуществляет сборку всех шейдеров в одну библиотеку, для единого доступа ко всем шейдерам	Обработка	Подсистема сборки библиотеки
3	Модуль синтак- сического ана- лиза	Осуществляет проверку программного кода шейдеров на ошибки	Обработка	Подсистема сборки библиотеки
4	Модуль опти- мизации	Осуществляет замены арифметических операций на аналогичные SIMD операции, оптимизирует участки кода и удаляет лишние	Обработка	Подсистема сборки библиотеки
5	Модуль взаимо- действия с бу- ферами	Осуществляет доступ к буферам моделей: полигонам, нормальям (работает многопоточно)	Хранение	Подсистема хранения данных
6	Модуль взаимо- действия с тек- стурами	Осуществляет доступ к текстурам моделей (работает многопоточно)	Хранение	Подсистема хранения данных
7	Модуль взаимо- действия с Pipeline	Осуществляет доступ к Pipeline: специальному упорядоченному множеству команд для вызова шейдеров (работает многопоточно)	Хранение	Подсистема хранения данных
8	Модуль взаимо- действия с шей- дерами	Осуществляет доступ к шейдерам для выполнения команд (работает многопоточно)	Хранение	Подсистема хранения данных

9	Среда передачи статических данных	Осуществляет передачу данных из подсистемы хранения данных к другим модулям и обратно (работает многопоточно)	Передача	
10	Модуль очереди команд	Осуществляет организацию правильного порядка команд (согласно переданным), для передачи в подсистему вычислений. Получать информацию может многопоточно с нескольких буферов	Передача	Подсистема исполнения команд
11	Модуль буферов команд	Осуществляет заполнение буферов команд для передачи в очередь команд, может выполняться многопоточно, тем самым, не позволив процессору ожидать выполнение текущей задачи	Передача	Подсистема исполнения команд
12	Модуль классификации команд	Осуществляет классификацию команд, полученных их Pipeline, разделяя на два типа шейдеров	Обработка	Подсистема исполнения команд
13	Модуль исполнения Вычислительных шейдеров	Осуществляет многопоточное вычисление переданной задачи, сохраняя результат в uniforms	Обработка	Подсистема вычислений
14	Модуль исполнения Вершинных, Геометрических и Фрагментных шейдеров	Осуществляет прохождение данных, из переданных буферов и uniforms, через графический конвейер: вершинный шейдер -> геометрический -> фрагментный	Обработка	Подсистема вычислений

15	Модуль хранения Frame	Осуществляет хранение полученных результатов из модуля 14 для последующей отрисовки на дисплее	Хранение	
16	Модуль вывода на дисплей	Осуществляет вывод frame	Вывод	
17	Модуль создания Pipeline	Осуществляет организацию порядка команд обработки буферов и uniforms, согласно пользовательским предпочтениям	Ввод	Подсистема инициализации моделей
18	Модуль создания буфера	Осуществляет выделение необходимого количества памяти на GPU для хранения данных	Ввод	Подсистема инициализации моделей
19	Модуль оптимизации текстуры	Осуществляет оптимизацию текстур, убирая лишние избыточную информацию из них (обычно, уменьшением кол-ва полигонов)	Обработка	Подсистема инициализации моделей
20	Модуль создания текстуры	Осуществляет выделение необходимого количества памяти на GPU для хранения данных	Ввод	Подсистема инициализации моделей
21	Модуль создание модели	Осуществляет ввод целой модели (например, в формате .obj), без предварительной подготовки	Ввод	Подсистема инициализации моделей
22	Среда передачи динамических данных (uniforms)	Осуществляет общий блок памяти для быстрой передачи изменяемых данных между CPU и GPU	Передача	
23	Устройство (обертка GPU)	Осуществляет интерфейс GPU	Ввод, Вывод	

Таблица 1.5. Описание связей элементов системы Metal

Номер связи на схеме	Название связи	Характеристика передаваемых данных
1	Получение данных Uniforms с GPU	Получение данных из общего буфера между CPU и GPU
2	Передача данных Uniforms с CPU	Передача данных в общий буфер между CPU и GPU
3	Передача команд для выполнения текущего задания	Передача в виде Pipeline: упорядоченного набора команд
4	Передача кадра для вывода на дисплей	Передача матрицы изображения
5	Передача упорядоченного набора команд	Очередь массива ссылок на шейдеры
6	Передача упакованной модели	Считанная с жесткого диска модель (например .obj)
7	Передача предобработанной текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек
8	Передача предобработанного буфера	Полигоны, нормали
9	Передача распакованного буфера	Полигоны, нормали
10	Передача распакованной текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек
11	Передача текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек
12	Сохранение оптимизированной текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек (точек не больше, чем изначально)
13	Сохранение буфера	Полигоны, нормали
14	Сохранение Pipeline	Передача в виде Pipeline: упорядоченного набора команд
15	Запись шейдера	Программный код на языке Metal Shading Language

16	Получение шейдера	Ссылка на функции из программного кода на языке Metal Shading Language
17	Запись Pipeline	Передача в виде Pipeline: упорядоченного набора команд
18	Получение Pipeline	Ссылка на pipeline
19	Запись текстуры	Передача матрицы изображения и привязанных к конкретным местам текстуры точек
20	Получение текстуры	Получение ссылки на матрицу изображения и привязанных к конкретным местам текстуры точек
21	Запись буфера	Полигоны, нормали
22	Получение буфера	Ссылки на структуры с полигонами, нормальями
23	Сохранение шейдеров	Программный код на языке Metal Shading Language
24	Запрос необходимых данных для выполнения текущей команды отрисовки	Ссылки на структуры с полигонами, нормальями и текстурами
25	Запрос необходимых данных для выполнения текущей команды вычисления	Ссылки на структуры с полигонами, нормальями и текстурами
26	Отправка команды для отрисовки	Ссылка на шейдер
27	Отправка команды для вычисления	Ссылка на шейдер
28	Отправка команды в очередь	Ссылка на шейдер
29	Отправка команд параллельно в буферы	Ссылка на шейдер
30	Запрос на uniforms	Разнородные данные из общего буфера между CPU и GPU
31	Отправка вычисленных uniforms	Разнородные данные из общего буфера между CPU и GPU
32	Запрос на uniforms	Разнородные данные из общего буфера между CPU и GPU
33	Передача отрисованного кадра в хранилище кадров	Передача матрицы изображения
34	Сигнал об ошибке	Исключение с указанием места в программном коде, где произошла ошибка

35	Сигнал об ошибке	Исключение с указанием места в программном коде, где произошла ошибка
36	Передача верного программного кода шейдера	Программный код на языке Metal Shading Language
37	Передача верного и оптимизированного программного кода	Программный код на языке Metal Shading Language

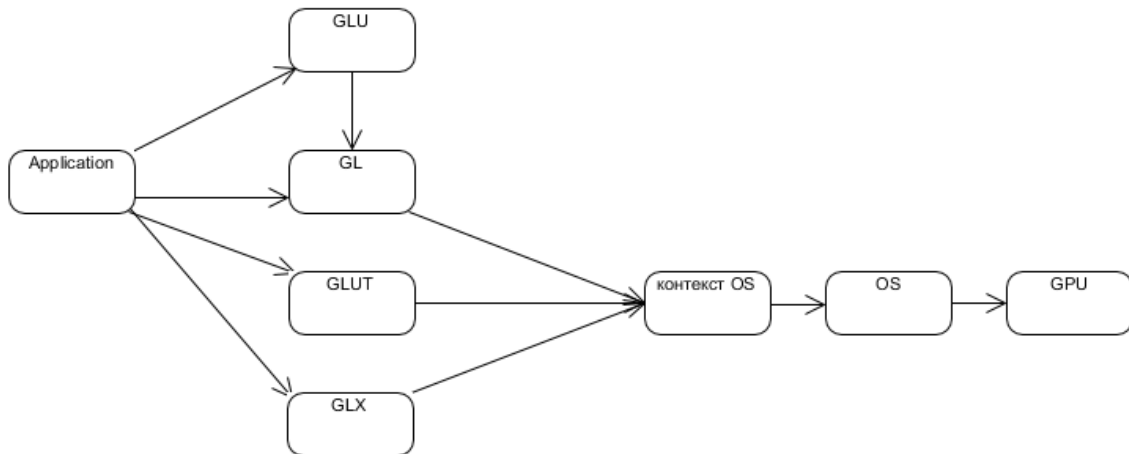


Рис. 1.3. Архитектура OpenGL на уровне приложений



Рис. 1.4. Архитектура Metal на уровне приложений

Таблица 1.6. Матрица смежности элементов системы Metal

Преимущества Metal над OpenGL:

- Изменения состояний теперь организованы в объекты состояний, которые могут просто переключаться, в то время как создание этих объектов будет обходиться дороже.
- Буферы команд теперь открыты и приложение может заполнять эти буферы и отправлять их в очередь команд, которая будет выполнять эти буферы в заданном порядке на GPU — таким образом приложение будет иметь полный контроль над заданием.
- В контексте macOS Metal имеет преимущества и ускорения (например, SIMD операции, упрощения текстур) за счет архитектуры библиотеки, построенной специально для платформ компании Apple

Недостатки Metal перед OpenGL:

- Не является кроссплатформенным, в отличие от OpenGL. Предназначен только для платформ Apple.

Выводы

В лабораторной работе отражены основные отличия двух популярных графических библиотек: OpenGL и Metal. В силу того, что Metal - платформозависимая библиотека, она имеет больше преимуществ в тех платформах, для которых она сделана (в этой работе - macOS). Это и SIMD операции, и упрощение текстур, и больший контроль над операциями. Metal позволяет разработчику самим решать, в какой очереди будут выполняться его шейдеры. Это ярко видно на подсистемах «Исполнение команд» и «Инициализация моделей».

Если перед разработчиком стоит вопрос о том, что выбрать для разработки приложения с графическим содержанием под macOS - выбор очевиден, это Metal. Но если задача состоит в разработке кроссплатформенного приложения, то лучше использовать 2 библиотеки под разные платформы (например, под Windows – OpenGL, под macOS – Metal), тем более, что синтаксис языков шейдеров в конкретных библиотеках практически не отличается.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение структуры системы.
2. Перечислите особенности построения модели структуры.
3. Какие элементы выделяются при моделировании информационных систем на уровне программно-технической архитектуры приложения?
4. Какие связи выделяются при моделировании информационных систем на уровне программно-технической архитектуры приложения?

Лабораторная работа №2. ОЦЕНКА СТРУКТУРНОЙ СЛОЖНОСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

ЦЕЛЬ РАБОТЫ

Научиться анализировать архитектуру существующей информационной системы и представлять ее в виде модели структуры.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Оценка структурной сложности информационной системы – важный инструмент для прогнозирования трудоемкости ее разработки и тестирования, оценки характеристик ее надежности.

В лабораторной работе рассматриваемые критерии структурной сложности интерпретируются в одном из аспектов – для оценки трудоемкости тестирования разрабатываемой ИС. При выполнении лабораторной работы необходимо определить значения по этим критериям и найти для них интерпретацию в других аспектах жизненного цикла информационной системы.

Для исследования структурной сложности структура системы ассоциируется с понятием «ориентированный взвешенный граф». Сложность графа, очевидно, зависит от числа вершин графа, числа его дуг и, возможно, от числа контуров. Становится возможным использование традиционных представлений теории графов для оценки структурной сложности.

Расчет сложности производится в первую очередь для замкнутых систем. В противоположность открытым системам они, как правило, наблюдаемы, а их сложность измерима, т.е. ее можно представить количественно.

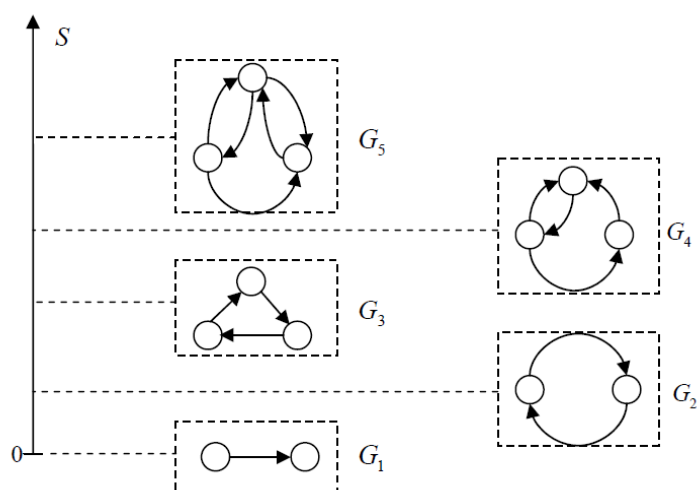


Рис. 2.1. Ранжировка типичных графовых структур по шкале структурной сложности

Структурная сложность орграфа G – это критерий $S(G)$, посредством которого орграфу G в однозначное соответствие ставится целое неотрицательное число $S(G)$, причем такое, что чем больше $S(G)$, тем граф G сложнее (рис. 2.1). Нулевая сложность соответствует понятию «простой граф» (G_1).

Структурную сложность орграфа можно представить в виде простейших количественных характеристик – критериев.

Первый критерий – число дуг в графе:

$$S^{(1)}(G)=m,$$

где m – это число дуг графа. Для описания этого числа можно использовать матрицу инциденций.

Применение критерия: В качестве интерпретации значения по этому критерию можно привести количество простых интеграционных тестов системы – тестов, проверяющих попарную интеграцию компонентов. Однако этот критерий не позволяет учесть структуру графа при расчете сложности.

Второй критерий – количество и состав бикомпонентов графа,

$$S^{(2)}(G)=\{v_{i,j}\}.$$

Для введения понятия бикомпонента графа вспомним терминологию:

- Связный неориентированный граф – для любых двух вершин существует соединяющий их маршрут.
- Связный орграф (б) – для любой пары вершин u, v либо u достижима из v , либо наоборот.
- Сильно связный орграф (а) – любые две вершины достижимы друг из друга.
- Слабо связный орграф (в) – не является связным, но при замене всех дуг на ребра порождает связный неориентированный граф.

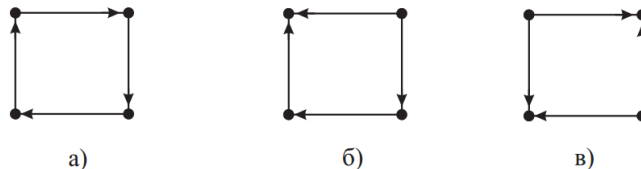


Рис. 2.2. К понятию связности графа

Тогда бикомпонент (компонент сильной связности) – это максимальный по включению сильно связанный подграф графа. Пример построения бикомпонента графа приведен на рис. 2.3. Орграф G_3 содержит 3 бикомпоненты (сильно связные подграфы максимальной размерности): $\{v_{1..3}\}$, $\{v_{4..7}\}$ и $\{v_{8..9}\}$.

Матрица достижимости P (рис. 2.3) отражает возможности достижения вершин в орграфе по путям, слагаемых из смежных дуг. Каждой бикомпоненте в матрице достижимости P_3 соответствует свой уникальный набор строк. Таким образом, по матрице достижимости можно выявить бикомпоненты орграфа.

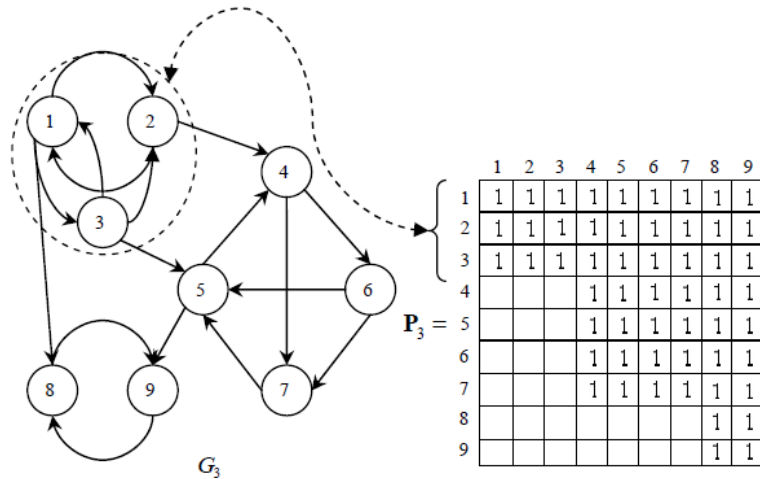


Рис. 2.3. Бикомпонент графа и его место в матрице достижимости

Очевидно, что любой граф можно рассматривать как набор бикомпонент, соединенных ребрами, не входящими ни в одну из его бикомпонент, при этом все контуры графа входят в бикомпоненты. Очевидно также, что, если входное воздействие попадает в одну из вершин бикомпоненты, то оно беспрепятственно распространится по всем другим ее вершинам. Поэтому с точки зрения прохождения информации через граф (а) можно рассматривать каждую бикомпоненту как одну вершину. Такое преобразование (б) называется графом Герца G^* (другое название – граф конденсаций). Пример построения графа Герца приведен на рис. 2.4.

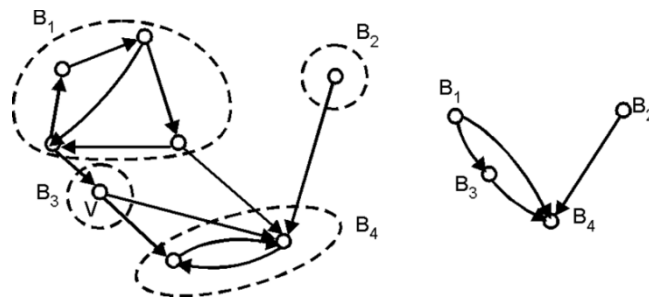


Рис. 2.4. Граф Герца для орграфа

Применение критерия: Количество выделенных бикомпонент может быть интерпретировано как количество подсистем, разработку и тестирование которых имеет смысл выполнять независимо с последующей интеграцией. Однако сравнительную оценку сложности каждой бикомпоненты этот критерий не дает.

Третий критерий сложности графов – число K элементарных контуров:

$$S^{(3)}(G) = K.$$

Контур — это конечный путь M , у которого начальная и конечная вершина совпадают. Контур называется элементарным, если все его вершины различны (за исключением начальной и конечной вершины).

Для определения количества контуров необходимо построить из исходного графа остовное дерево. *Остовное дерево (остов)* — это подграф данного графа, содержащий все его вершины и являющийся деревом. Рёбра графа, не входящие в остов, называются **хордами** графа относительно остова. Выбор остовного дерева графа неоднозначен, для одного и того же графа их может быть несколько. На рис. 2.5, а представлен пример связного графа, а на рис. 2.5, б и рис. 2.5, в - его остовные деревья. Для дерева 2.5, б, ветви — это ребра б, г, е, ж, и, хорды — ребра а, в, д, к; для дерева рис. 2.5, в, ветви — это ребра а, б, в, д, и, хорды — г, е, ж, к.

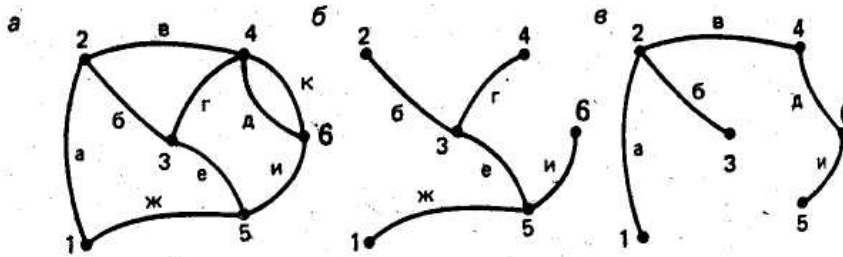


Рис. 2.5. Граф (а) и его остовные деревья (б, в).

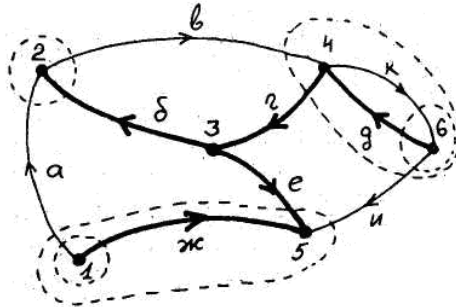


Рис. 2.6. Орграф, соответствующий графу рис. 2.5, а.

Соотношение ветвей и хорд в графе описывается с помощью матрицы контуров. Количество столбцов матрицы контуров равно числу ветвей выбранного для графа остовного дерева, а количество строк - числу хорд. В случае орграфа при построении матрицы контуров нужно учитывать направление ребер, входящих в контур, а именно: каждая хорда графа поочередно включается в остовное дерево, при этом образуется замкнутый контур. Обход этого контура выполняется в направлении, заданном направлением хорды; в строке матрицы, соответствующей данной хорде, ставится +1, если направление ветви дерева совпадает с направлением обхода контура, -1, если направление ветви дерева противоположно, 0, если ветвь не входит в данный контур.

Получение матрицы контуров для графа, показанного на рис. 2.5, а, представлено в табл. 2.1.

Таблица 2.1. Матрица контуров

	<i>б</i>	<i>г</i>	<i>д</i>	<i>е</i>	<i>ж</i>
<i>а</i>	-1	0	0	+1	-1
<i>в</i>	+1	+1	0	0	0
<i>к</i>	0	0	+1	0	0
<i>и</i>	0	-1	-1	-1	0

При подключении хорды *а* образуется контур из ветвей дерева *б*, *е*, *ж*, в столбцах матрицы, соответствующих ветвям *б* и *ж*, появится -1, в столбце, соответствующем ветви *е*, будет +1, остальные столбцы содержат 0. Аналогично заполняются и другие строки.

Применение критерия: Практически любая связь компонентов информационной системы работает по принципу «посыл-отклик». Количество контуров может определить количество соответствующих тестов.

Однако любые кольца (независимо от числа вершин в них) в соответствии с этим критерием одинаково сложны. В то же время очевидно, что любое увеличение длины контура значительно увеличивает сложность локализации ошибки при тестировании: при большем количестве элементов и связей в контуре обнаружить несоответствие отклика посылу мы, как правило, можем только в одной точке для всего контура целиком, но это не дает нам информации о том, где именно дефект.

Поэтому большой интерес представляет оценка приоритетности дуг. Дуга *di* является более приоритетной, чем *dj*, если ее удаление приводит к более существенному уменьшению структурной сложности. В замкнутой системе, имеющей структуру сильно связного орграфа, дуга *di* будет более приоритетной, чем *dj*, если

- 1) ее вес γ_i минимален в D ;
- 2) через дугу *di* проходит максимальное число контуров.

Примером может служить орграф типа *гамак* – орграф с критической дугой, через которую проходят сугубо все контуры. Если произвести разрыв критической дуги, получим орграф без контуров. Пример гамака показан на рис. 2.7. Критическая дуга ($1 \rightarrow 2$) разрывает все контуры, и она – наиболее приоритетная, если $\forall i \neq j: \gamma_i \cong \gamma_j; i, j = 1, 17$. Однако если предположить, что веса дуг существенно различны, $\gamma_1 \gg \gamma_i, i = 2, 17$, то наиболее приоритетной дугой может оказаться отнюдь не критическая дуга.

Поэтому дуги, входящие в граф системы, сортируются по следующим признакам (перечислены в последовательности проверки):

- 1) числу контуров, проходящих через дугу (по убыванию);
- 2) весу дуги (по возрастанию);
- 3) индексу начала дуги (по возрастанию);
- 4) индексу конца дуги (по возрастанию).

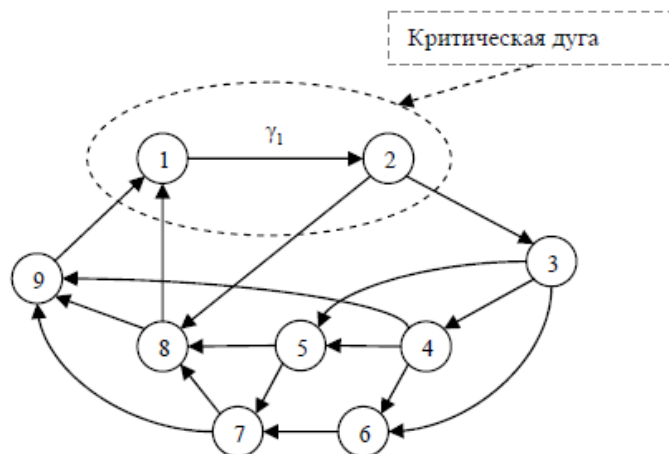


Рис. 2.7. Гамак с критической дугой

При таком упорядочении наиболее приоритетной окажется дуга, входящая в максимальное число контуров, и, если таких дуг несколько, то более приоритетной будет дуга, имеющая меньший вес. Наоборот, наименее приоритетной окажется дуга, входящая в минимальное число контуров, и, если таких дуг несколько – дуга с максимальным весом.

Сортировка дуг сопровождается перестановкой столбцов матрицы контуров:

- 1) столбцы матрицы контуров группируются по убыванию суммарного количества единичных элементов в этих столбцах;
- 2) если возникли подматрицы из столбцов с одинаковым количеством единиц, расставляем столбцы по убыванию веса соответствующих им дуг;
- 3) если в одной подматрице находятся дуги с одинаковым весом и одинаковым числом единиц в соответствующих столбцах, они расставляются в лексикографическом порядке по индексам дуг.

В общем случае сложность невзвешенного орграфа можно оценить как произведение числа дуг на число контуров. Взвешенный орграф представляет собой совокупность вершин, дуг и их весов. Эта совокупность представлена двумя матрицами – взвешенными матрицами смежности (X) и инцидентности (B). Контуров представлены взвешенной матрицей контуров (C).

Таким образом, чтобы оценить сложность взвешенного орграфа, моделирующего конкретную систему, в общем случае необходимо:

- найти произведение матриц: взвешенная матрица смежности умножается на взвешенную матрицу инцидентности, а затем полученное произведение – на транспонированную взвешенную матрицу контуров (C^*);
- произвести спектральное разложение полученной матрицы по собственным числам;
- полученный спектр матрицы может использоваться для сравнительной оценки сложности сопоставляемых систем.

Выполнение указанной программы связано с большими вычислительными сложностями; с другой стороны, редко удается задать веса дуг графа системы с адекватной точностью. Поэтому на практике часто ограничиваются ориентировочными оценками.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

1. Определить число дуг в графе системы, выбранной в ЛР1, используя матрицу инцидентности, построенную в ЛР 1.

2. Определить количество и структуру бикомпонент для выбранной системы. Построить для нее граф Герца. Дать содержательное описание построенных артефактов.

3. Построить матрицу контуров для выбранной системы, дать ее содержательную интерпретацию.

4. Перестроить полученную матрицу контуров, отсортировав ее строки по убыванию количества связей, входящих в контур. Дать содержательное описание назначения каждого контура в информационной системе, инициатора (инициаторов) и конечных потребителей информации в каждом контуре. Определить, являются ли однородными отдельные связи, входящие в контур (используют одинаковый способ связи, одинаковый или просто преобразуемый формат данных и т.п.).

Перестроить полученную матрицу контуров, отсортировав ее столбцы по убыванию количества контуров в которые входит соответствующая связь. Дать содержательное описание различий в работе связи при ее использовании в различных контурах.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Матрицы смежности и инцидентности для графа системы OpenGL из ЛР1 представлены в табл. 2.2 и 2.3.

Анализ матриц показывает количество дуг – 27 дуг.

Матрица достижимости для графа системы OpenGL из ЛР1 представлена в табл. 2.4. Граф Герца, построенный в соответствии с матрицей достижимости, показан на рис. 2.8.

Из этой матрицы можно выделить две бикомпоненты:

- 2, 3, 4, 5 – это подсистема управления памятью с модулем статичных данных. Это вполне целостная подсистема, в которой каждый компонент общается с модулем передачи статичных данных.
- 6, 7, 8, 9, 10, 15, 16 – это связка подсистем исполнения шейдеров и конвейера команд. Это сильно взаимосвязанные подсистемы, поэтому они находятся в одной компоненте сильной связности. Отметим, что в целом

Таблица 2.4. Матрица достижимости

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
7	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
8	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
9	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
10	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
11	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0
14	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
16	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
17	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
18	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

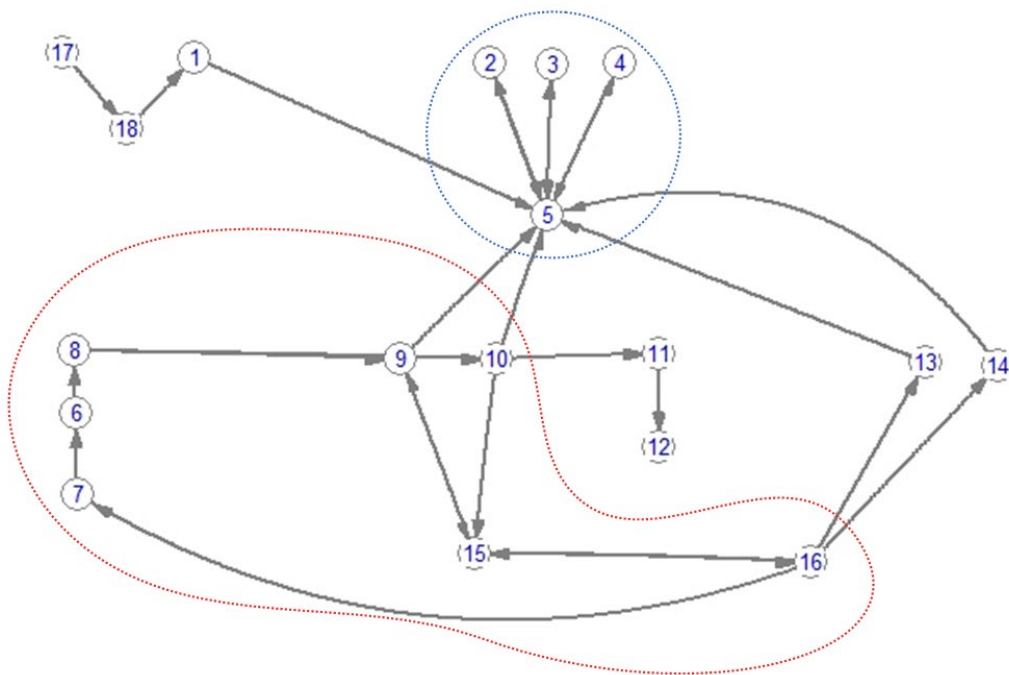


Рис. 2.8. Граф Герца

Остовное дерево на основе графа системы показано на рис. 2.9. Жирным цветом выделены ребра, которые входят в дерево. Соответствующая матрица контуров представлена в табл. 2.5.

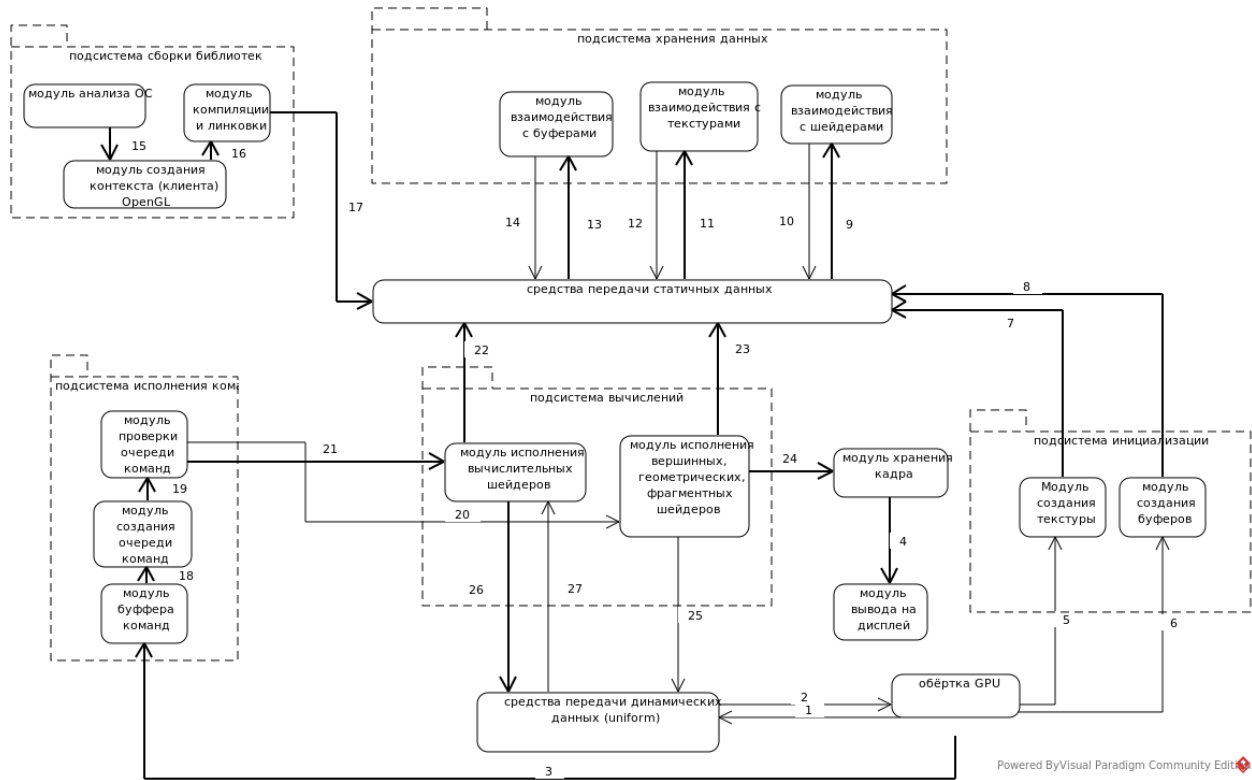


Рис. 2.9. Остовное дерево

Таблица 2.5. Матрица контуров

№	3	7	8	9	11	13	18	19	21	22	23	26
1	-1						-1	-1	-1			-1
2	+1						+1	+1	+1			+1
5	-1	+1					-1	-1	-1	-1		
6	-1		+1				-1	-1	-1	-1		
10				+1								
12					+1							
14						+1						
20									-1	-1	+1	
25										+1	-1	-1

В построенной таблице присутствует очень мало контуров, состоящих из двух элементов. Большую часть составляют сложные контуры, что говорит о сложности системы и ее тестирования.

Результат первой сортировки таблицы контуров (строки по убыванию количества связей, входящих в контур) представлен в табл. 2.6.

Таблица 2.6. Матрица контуров (первая сортировка)

№	3	7	8	9	11	13	18	19	21	22	23	26
6	-1		+1				-1	-1	-1	-1		
5	-1	+1					-1	-1	-1	-1		
2	+1						+1	+1	+1			+1
1	-1						-1	-1	-1			-1
20									-1	-1	+1	
25										+1	-1	-1
10				+1								
14						+1						
12					+1							

Контур 6 работает с передачей и обработкой буферов данных. Положительная часть этого контура отвечает за передачу данных, отрицательная – за обработку данных. Аналогично с контуром 6 работает контур 5, только в последнем вместе буферов находятся текстуры. Контур 2 показывает цикл исполнения вычислительных шейдеров, причем в правильном порядке. В этом контуре на входе – команды, на выходе – переподготовленные просчитанные данные. Контур 1 показывает такой же цикл, как и контур 2, но наоборот. Этот контур говорит лишь о том, что есть некий модуль, с которым существует обратная связь (т.е. средства передачи данных). Контур 20 при таком построении основного дерева не несет смысловой нагрузки. Контур 25 показывает, как гипотетически можно передавать информацию между шейдерами, но на практике такое не используется из-за “дорогой” синхронизации потоков. Контур 14, 12, 10 показывают взаимодействие с модулями подсистемы хранения данных.

Результат второй сортировки таблицы контуров (столбцы по убыванию количества контуров, в которые входит соответствующая связь) представлены в табл. 2.7.

Узлы 21, 3, 18, 19 соответствуют сценарию, когда необходимо передать команды для исполнения на GPU. Так как с этим сценарием связаны практически все процессы на GPU, то эти узлы находятся в большинстве контуров. Узлы 22 и 26 необходимы для взаимодействия вычислительных шейдеров и данных, которые необходимо обрабатывать. Такое количество контуров, в которые входят эти узлы, обусловлено построением основного дерева. В общем случае они могут быть задействованы в 1-2 контурах. Узел

23 необходим для взаимодействия графических шейдеров и статичных данных. В этой схеме он имеет один реальный контур и один теоретический. Узлы 7, 8, 9, 11, 13 выполняют в каждом контуре свою единственную задачу – передать и сохранить данные в подсистеме хранения данных.

Таблица 2.7. Матрица контуров (вторая сортировка)

№	21	3	18	19	22	26	23	8	7	9	13	11
6	-1	-1	-1	-1	-1			+1				
5	-1	-1	-1	-1	-1				+1			
2	+1	+1	+1	+1		+1						
1	-1	-1	-1	-1		-1						
20	-1				-1		+1					
25					+1	-1	-1					
10										+1		
14											+1	
12												+1

Выводы

1. Удалось достаточно точно определить подсистемы библиотеки, но в силу слабой связности некоторых из них часть систем вошла в подсистемы других подсистем
2. Самое нагруженное место системы OpenGL – это подсистема исполнения команд, так как через нее проходит большинство контуров.
3. Самые используемые подсистемы – подсистемы исполнения команд и вычисления шейдеров.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение структурной сложности орграфа
2. Перечислите критерии структурной сложности графа
3. Дайте определение бикомпоненты графа
4. Дайте определение матрицы достижимости
5. Дайте определение графа Герца
6. По каким признакам сортируются дуги, входящие в граф системы?

Лабораторная работа №3. ОЦЕНКА ПРОЕКТНЫХ ХАРАКТЕРИСТИК АРХИТЕКТУРНОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

ЦЕЛЬ РАБОТЫ

Научиться оценивать проектные характеристики архитектурной модели информационной системы.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Архитектурные модели ПО имеют следующие классические проектные характеристики: модульность, информационная закрытость, связность, сцепление, сложность.

Модульность – свойство системы, которая может подвергаться декомпозиции на ряд внутренне связанных и слабо зависящих друг от друга модулей.

Выбор оптимального количества модулей приводит к минимальной стоимости разработки ПО. Для оценки оптимального числа модулей существуют только требования:

- Интерфейсы модуля должны быть проще, чем его внутренняя организация;
- Эффективнее использовать готовый модуль, чем реализовывать его функциональность в другом модуле.

Информационная закрытость означает, что ПО удовлетворяет следующим требованиям:

- все модули независимы, обмениваются только информацией, необходимой для работы;
- доступ к операциям и структурам данных модуля ограничен.

Для оценки информационной закрытости необходимо рассматривать два типа связей – внутренние и внешние. Соответственно, появляются две подхарактеристики – связность внутри модуля и сцепление между модулями.

Связность модуля (Cohesion) – это мера зависимости его частей. Связность – внутренняя характеристика модуля. Чем выше связность модуля, тем лучше результат проектирования.

Для измерения связности используют метрику силы связности (СС). Чем выше значение СС, тем лучше. Существует 7 типов связности, для которых значения СС определены экспертным путем.

1. Связность по совпадению (СС=0). В модуле отсутствуют явно выраженные внутренние связи, например:

Модуль Разные функции (какие-то параметры)

поздравить с Новым годом (...)
 проверить исправность аппаратуры (...)
 заполнить анкету героя (...)
 измерить температуру (...)

2. Логическая связность (СС=1). Части модуля объединены по принципу функционального подобия. Например, модуль состоит из разных подпрограмм обработки ошибок.

3. Временная связность (СС=3). Части модуля не связаны, но необходимы в один и тот же период работы системы. Пример – модуль инициализации компьютера.

4. Процедурная связность (СС=5). Части модуля связаны порядком выполняемых ими действий, реализующих некоторый сценарий поведения. Например:

Модуль Вычисление средних значений
 используется Таблица-А, Таблица-В
 вычислить среднее по Таблица-А
 вычислить среднее по Таблица-В
 вернуть среднееТабл-А, среднееТабл-В

5. Коммуникативная связность (СС=7). Части модуля связаны по данным (работают с одной и той же структурой данных). Пример – Модуль «Отчет и средняя зарплата».

6. Информационная (последовательная) связность (СС=9). Выходные данные одной части используются как входные данные в другой части модуля.

7. Функциональная связность (СС=10). Модуль содержит элементы, участвующие в выполнении одной и только одной, проблемной функции (единой с точки зрения клиента) – например, вычислить зарплату сотрудника.

Возможны более сложные случаи, когда с модулем ассоциируются несколько уровней связности. В этих случаях следует применять одно из двух правил:

- правило параллельной цепи. Если все действия модуля имеют несколько уровней связности, то модулю присваивают самый сильный уровень связности;
- правило последовательной цепи. Если действия в модуле имеют разные уровни связности, то модулю присваивают самый слабый уровень связности.

Например, модуль может содержать некоторые действия, которые связаны процедурно, а также другие действия, связанные по совпадению. В этом случае применяют правило последовательной цепи и в целом модуль считают связным по совпадению.

Сцепление модулей (Coupling) – мера взаимозависимости модулей по данным. Сцепление – внешняя характеристика модуля, которую желательно уменьшать.

Количественно сцепление измеряется метрикой «степень сцепления» (СЦ). Чем ниже СЦ, тем лучше. Выделяют 6 типов сцепления, для которых значение СЦ определяется экспертно.

1. Сцепление по данным (СЦ=1). Модуль А вызывает модуль В. Все входные и выходные параметры вызываемого модуля — простые элементы данных (рис. 3.1).



Рис. 3.1. Сцепление по данным

2. Сцепление по образцу (СЦ=3). В качестве параметров используются структуры данных (рис. 3.2).

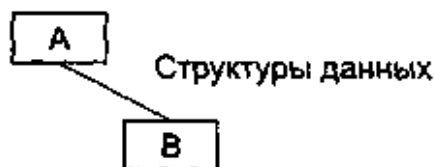


Рис. 3.2. Сцепление по образцу

3. Сцепление по управлению (СЦ=4). Модуль А явно управляет функционированием модуля В (с помощью флагов или переключателей), посылая ему управляющие данные (рис. 3.3).

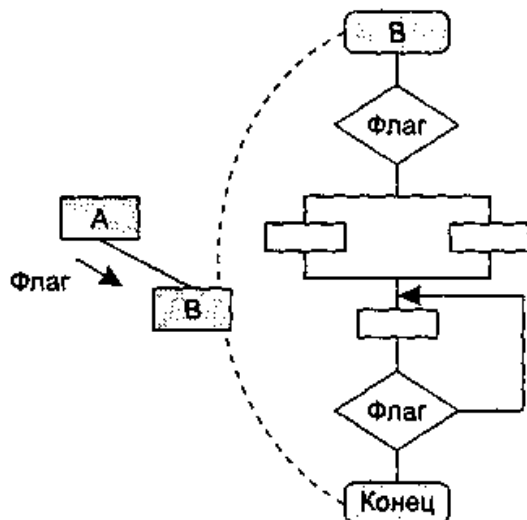


Рис. 3.3. Сцепление по управлению

4. Сцепление по внешним ссылкам (СЦ=5). Модули А и В ссылаются на один и тот же глобальный элемент данных.

5. Сцепление по общей области (СЦ=7). Модули разделяют одну и ту же глобальную структуру данных (рис. 3.4).

6. Сцепление по содержанию (СЦ=9). Один модуль прямо ссылается на содержание другого модуля (не через его точку входа). Например, коды их команд перемежаются друг с другом (рис. 4).

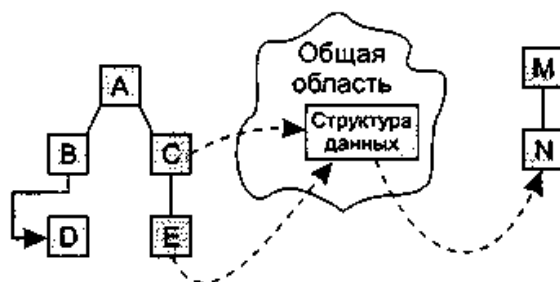


Рис. 3.4. Сцепление по общей области и содержанию

На рис. 4 видно, что модули В и D сцеплены по содержанию, а модули С, Е и N сцеплены по общей области.

Каждый из выделенных модулей можно рассматривать как вершину графа – модели более высокого уровня абстракции. Для него правомерно применить *топологическую сортировку* – найти перестановку вершин (топологический порядок), соответствующую порядку, задаваемому всеми рёбрами графа. То есть требуется перенумеровать его вершины таким образом, чтобы каждое ребро вело из вершины с меньшим номером в вершину с большим.

Топологическая сортировка может быть не единственной (например, если граф – пустой; или если есть три такие вершины a, b, c , что из a есть пути в b и в c , но ни из b в c , ни из c в b добраться нельзя). Топологической сортировки может не существовать вовсе – если граф содержит циклы (поскольку при этом возникает противоречие: есть путь и из одной вершины в другую, и наоборот).

В общем случае при помощи топологической сортировки строится корректная последовательность выполнения действий, всякое из которых может зависеть от другого: последовательность прохождения учебных курсов студентами, установки программ при помощи пакетного менеджера, сборки исходных текстов программ при помощи Makefile'ов.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

Взяв за основу программно-техническую архитектуру ИС, построенную в предыдущих лабораторных работах, выделить модули в соответствии с обозначенными выше требованиями. Полученная структура может как совпадать с уже построенным ранее графом, так и отличаться от него.

1. Для каждого модуля определить степень внутренней связности по предложенной классификации.

2. Для каждой связи определить тип сцепления по предложенной классификации.

3. Для полученной структуры модулей проверить на содержательном уровне выполнение требований к модульности и информационной закрытости. Построить диаграмму размещения модулей по физическим узлам для тех систем, где это возможно, или по изолированно выполняющимся процессам для всех остальных. За основу взять в первом случае требования к платформе и технологии реализации, а во втором случае – оценку взаимовлияния сбоев и критического повышения потребляемых ресурсов, приводящих к взаимной блокировке.

4. Выполнить топологическую сортировку построенных модулей и ответить на следующие вопросы:

- В каком порядке должны быть разработаны модули, чтобы каждый из них после разработки сразу можно было бы протестировать на полную функциональность (то есть появление потом новых модулей не привело бы к необходимости его перетестирования)?
- В каком порядке нужно принимать проектные решения по модулям (какие выбрать технологии, структуры данных, интерфейсы и т.п.), чтобы потом не менять эти решения в процессе проектирования остальных модулей?

Примечание. Если исходный граф модулей имеет циклы, перед осуществлением топологической сортировки его нужно преобразовать в остовное дерево, как это было описано в лабораторной работе №2.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Степень внутренней связности для модулей системы OpenGL показана в табл. 3.1, 3.2. Как видно из табл. 3.2, все модули оцениваются достаточно хорошими показателями СС, что говорит о хорошем качестве проектирования системы.

Степень сцепления для модулей системы OpenGL показана в табл. 3.3, 3.4. Как видно из табл. 3.4, сцепления модулей преимущественно происходят по образцу, что говорит о простоте интерфейсов модулей.

По результатам выполнения первых двух заданий можно говорить о том, что система соответствует оптимальным характеристикам архитектурной модели ПО:

1. Модульность – система строится из сильно связанных внутри модулей, имеющих простые интерфейсы

2. Информационная закрытость – каждый модуль владеет своими данными и общается только при необходимости путем обмена информацией с другими модулями.

Таблица 3.1. Обозначения для связности

Степень внутренней связности	Обозначение
По совпадению	СС0
Логическая	СС1
Временная	СС3
Процедурная	СС5
Коммуникативная	СС7
Информационная	СС9
Функциональная	СС10

Таблица 3.2. Оценки связности модулей

Но- мер	Название элемента	Степень внутрен- ней связности
1	Модуль сборки библиотек	СС9
2	Модуль взаимодействия с буферами	СС1
3	Модуль взаимодействия с текстурами	СС1
4	Модуль взаимодействия с шейдерами	СС1
5	Среда передачи статичных данных	СС3
6	Модуль исполнения команд	СС9
7	Модуль исполнения Вычислительных шейдеров	СС10
8	Модуль исполнения Вершинных, Геометрических и Фрагментных шейдеров	СС10
9	Модуль вывода на дисплей	СС5
10	Модуль создания буфера	СС7
11	Модуль создания текстуры	СС7
12	Среда передачи динамических данных (uniforms)	СС3
13	Устройство (обертка GPU)	СС3

Таблица 3.3. Обозначения для сцепления

Тип сцепления модулей	Обозначение
По данным	СЦ1
По образцу	СЦ3
По управлению	СЦ4
По внешним ссылкам	СЦ5
По общей области	СЦ7
По содержанию	СЦ9

Таблица 3.4. Оценки сцепления модулей

Номер	Название связи	Тип сцепления модулей
1	Получение данных Uniforms с GPU	СЦ3
2	Передача данных Uniforms с CPU	СЦ3
3	Передача команд для выполнения текущего задания	СЦ4
5	Передача текстуры	СЦ3
6	Передача предобработанного буфера	СЦ3
7	Сохранение текстуры	СЦ3
8	Запись шейдера	СЦ3
9	Получение шейдера	СЦ3
10	Получение буфера	СЦ3
11	Запись текстуры	СЦ3
12	Получение текстуры	СЦ3
13	Сохранение буфера	СЦ3
14	Запись буфера	СЦ3
17	Сохранение шейдеров	СЦ3
20	Отправка команды для отрисовки	СЦ4
21	Отправка команды для вычисления	СЦ4
22	Запрос необходимых данных для выполнения текущей команды отрисовки	СЦ3

23	Запрос необходимых данных для выполнения текущей команды вычисления	СЦЗ
24	Отправка вычисленных uniforms	СЦЗ
25	Запрос на uniforms	СЦЗ
26	Запрос на uniforms	СЦЗ
27	Получение данных Uniforms с GPU	СЦЗ

Диаграммы размещения модулей по изолированно выполняющимся процессам представлены на рис. 3.5, 3.6, 3.7.



Рис. 3.5. Диаграмма процесса инициализации и загрузки текстур

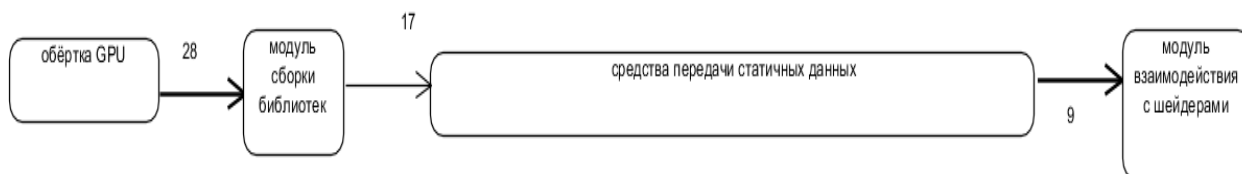


Рис. 3.6. Диаграмма процесса компиляции библиотек

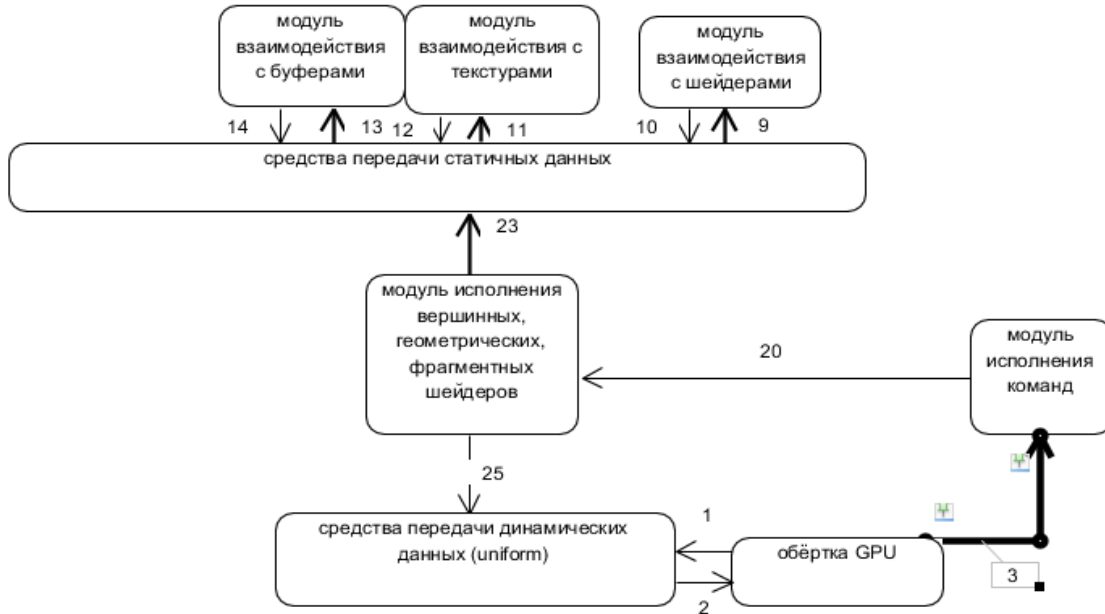


Рис. 3.7. Диаграмма процессов выполнения шейдеров

Анализ процессов показывает, что в системе OpenGL соблюдается информационная закрытость, в разных процессах участвуют только необходимые для данного процесса модули. Обмен данных также находится только на уровне информации, необходимой для работы.

Результат топологической сортировки выделенных модулей иллюстрируется на рис. 3.8.

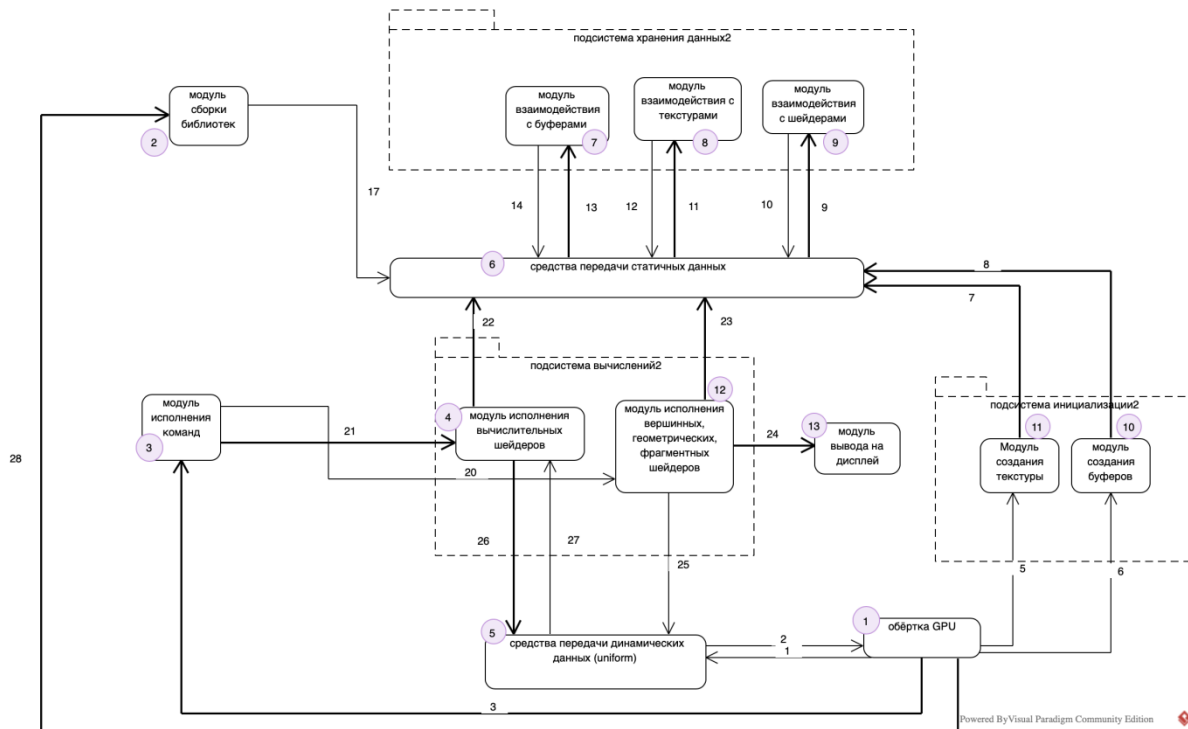


Рис. 3.8. Топологическая сортировка выделенных модулей

Рис. 3.8 показывает, что разработку модулей следует осуществлять с середины порядка модулей, согласно топологической сортировке: сначала реализовывать главную часть, а потом дошивать остальные модули.

Проектные же решения нужно стараться принимать от первоначальных (логических) узлов к последним. Нужно стремиться делать решение от общего, к частному, иначе можно написать модуль, сделать к нему тесты, а потом окажется, что он использует неэффективную структуру данных, которую можно было упростить на родительских узлах.

Выводы

В ходе работы оценены проектные характеристики архитектурной модели информационной системы OpenGL.

Выявлено, что система соответствует таким важным требованиям, как модульность и информационная закрытость.

Выделены изолированные процессы, для них построены диаграммы размещения.

Произведена топологическая сортировка остоного дерева, построенного на исходном графе системы. Выявлена правильная последовательность разработки каждого модуля.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение модульности как проектной характеристики программного обеспечения
2. Дайте определение информационной закрытости как проектной характеристики программного обеспечения
3. Дайте определение связности как проектной характеристики программного обеспечения
4. Дайте определение сцепления как проектной характеристики программного обеспечения
5. Дайте определение сложности как проектной характеристики программного обеспечения
6. С какой целью можно использовать топологическую сортировку при проектировании программного обеспечения?

Лабораторная работа №4. АНАЛИЗ ДИНАМИЧЕСКИХ ХАРАКТЕРИСТИК ИН- ФОРМАЦИОННОЙ СИСТЕМЫ НА ОСНОВЕ ЕЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

ЦЕЛЬ РАБОТЫ

Ознакомиться с методиками оценки динамических характеристик информационной системы (ИС) с применением сетей Петри.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1°. Основные определения. На базе созданной в предыдущих лабораторных работах структурной схемы (функциональной модели) ИС можно построить модель для оценивания динамических характеристики ИС. Для этого чаще всего используется аппарат сетей Петри.

Сеть Петри $PN = (P, T, F)$ определяется как двудольный граф, т.е. все вершины графа относятся к одному из двух классов – позициям P (изображаются окружностями) и переходам T (изображаются отрезками прямой). Дуги в сетях Петри – направленные, причем каждая дуга связывает вершины только разных классов. В позициях могут размещаться метки F (изображаются черными кружками), способные перемещаться по сети. Событием называют срабатывание перехода, при котором метки из входных позиций этого перехода перемещаются в выходные позиции.

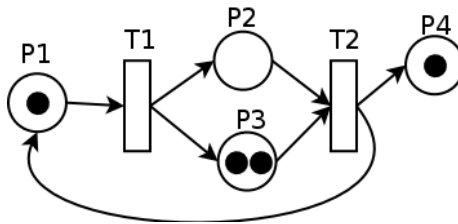


Рис. 4.1. Пример сети Петри

Для моделирования потоков работ (активностей в бизнес-процессах) или потоков операций в технологических процессах целесообразно использовать WF-сети – подкласс сетей Петри, называемый также сетями потоков работ. Сеть Петри $PN = (P, T, F)$ называется сетью потоков работ (WF-сетью), если выполняются следующие условия:

- существует только одна исходная позиция i , такая что отсутствуют переходы, входящие в i ;
- существует только одна конечная позиция o , такая, что отсутствуют переходы, выходящие из o ;
- каждый узел данной сети расположен на пути от i к o .

2°. Представление функциональной модели ИС в виде WF-сети.

Вершинами графа описываются операции, которые осуществляет ИС. Дугами описываются переходы между операциями, сопровождающиеся передачей данных от предыдущего этапа. Под передаваемыми данными могут пониматься как собственно данные для обработки на следующем этапе, так и как параметры управления следующим этапом.

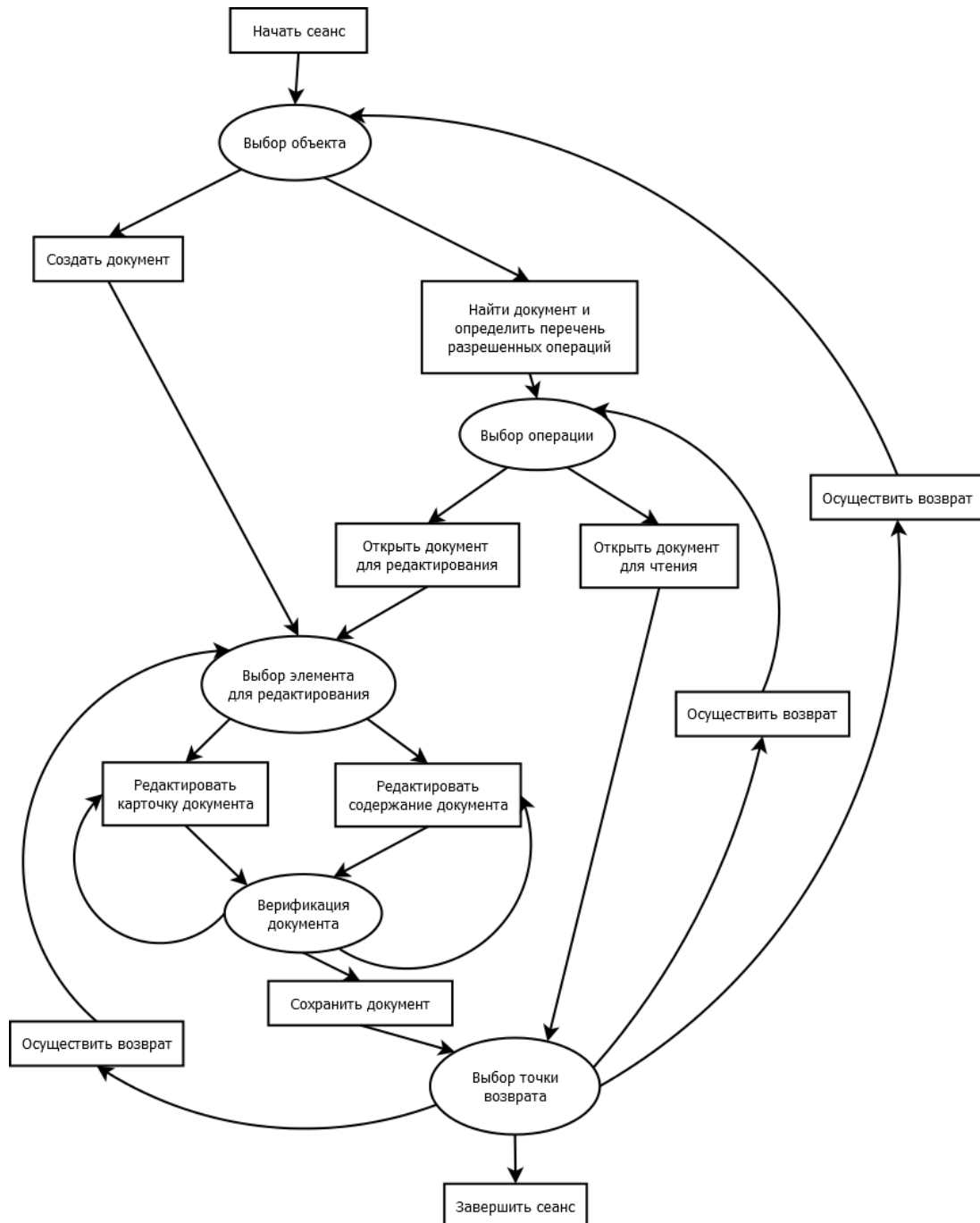


Рис. 4.2. Пример графа

Граф строится как двудольный, т.е. выделяются вершины двух типов:

1. Операция, реализующая осуществленное на одном из предыдущих этапов управление (реализация управления). Обозначены прямоугольниками. Такая операция может иметь **не более двух** выходов — выход в случае успешного и выход в случае неуспешного выполнения операции. Для упрощения модели допускается обозначать для таких операций только выход в случае успешного выполнения операции, если результатом неуспешного выполнения является завершение соответствующего процесса и переход к точке выхода.
2. Операция, определяющая дальнейший путь процесса (осуществление управления). Обозначены овалами. Такая операция должна иметь не менее двух выходов, приводящих к двум разным вершинам.

Граф имеет одну вершину, имеющую только выходящие дуги и не имеющую входящих дуг. Такую вершину будем называть точкой входа. Она обозначает начало работы с информационной системой. Аналогично существует одна вершина, которая имеет только входящие дуги и не имеет выходящих. Такую вершину будем называть точкой выхода. Она обозначает окончание работы с информационной системой.

3°. Проверка двудольности графа

Чтобы проверить граф на предмет двудольности, достаточно в каждой компоненте связности выбрать любую вершину и пометить оставшиеся вершины во время обхода графа (например, поиском в ширину или в глубину) поочередно как чётные и нечётные (см. рис. 2). Если при этом не возникнет конфликта, все чётные вершины образуют множество вершин одного типа (например, позиций), а все нечётные — множество вершин другого типа (переходов) соответственно.

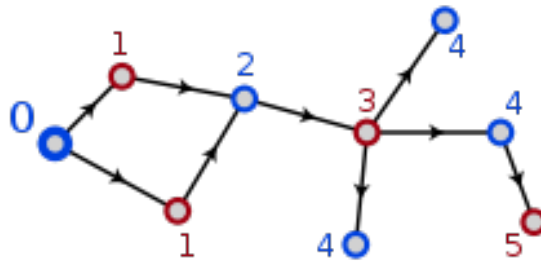


Рис. 4.3. Проверка двудольности с помощью чётности расстояний

4°. Анализ свойств ИС с применением WF-сети

4.1°. WF-сети используются для проверки графов потоков работ на отсутствие структурных конфликтов (бездефектность). Свойство бездефектности, или правильной завершаемости, соответствует следующим требованиям:

- конечная позиция о достижима при любой последовательности переходов от начальной позиции i ;
- WF-сеть не содержит лишних позиций (которые никогда не будут выполнены);

- при достижении конечной позиции i данной сети не должно оставаться меток в промежуточных позициях.

4.2°. WF-сети также используются для выявления ошибок абстрактного сценария ИС. Применительно к сценарию проверяются три свойства сети:

- сеть должна быть ограниченной. Позиция сети Петри ограничена (k -ограничена), если существует такое целое число k , что число объектов в этой позиции никогда не превышает k . Число k называют емкостью позиции. Сеть Петри ограничена, если ограничены все ее позиции;
- при работе сети не должны появляться неконечные тупиковые состояния, в которых не активирован ни один переход;
- при работе сети не должно возникать "ловушек" – циклов без выхода (объект может попасть в "ловушку", циклически циркулировать в ней, но не может выйти из "ловушки").

Сценарий бизнес-системы будем считать корректным, если:

- корректны все образующие его сценарные модули
- модули корректно согласованы.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

1. Рассматривая функциональную модель ИС, построенную в предыдущих лабораторных работах, как граф, построить на его основе WF-сеть в соответствии с требованиями пункта 2, руководствуясь следующими рекомендациями:
 - a. Заменить название каждого компонента системы, являющегося вершиной исходного графа, на название операции, которую выполняет этот компонент. При невозможности выделить одну операцию, рекомендуется разбить такой компонент на несколько, установив связи между ними.
 - b. Осуществить проверку двудольности получившегося графа, получив разделение на два типа вершин, указанных в пункте 2. При нарушении двудольности внести соответствующие изменения в схему.
 - c. Осуществить проверку на наличие и единственность входной и выходной вершин. При нарушении этого требования внести соответствующие изменения в схему.
2. Составить таблицы, характеризующую вершины обоих типов.
3. Выполнить анализ свойств ИС с применением WF-сети (пп. 4.1° и 4.2°). Для выполнения п. 4.2° составить один типичный сценарий использования ИС. Дать содержательную интерпретацию полученных результатов.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

WF-сеть, построенная по системе OpenGL, представлена на рис. 4.4 в полном виде и на рис. 4.5 – в упрощенных обозначениях.

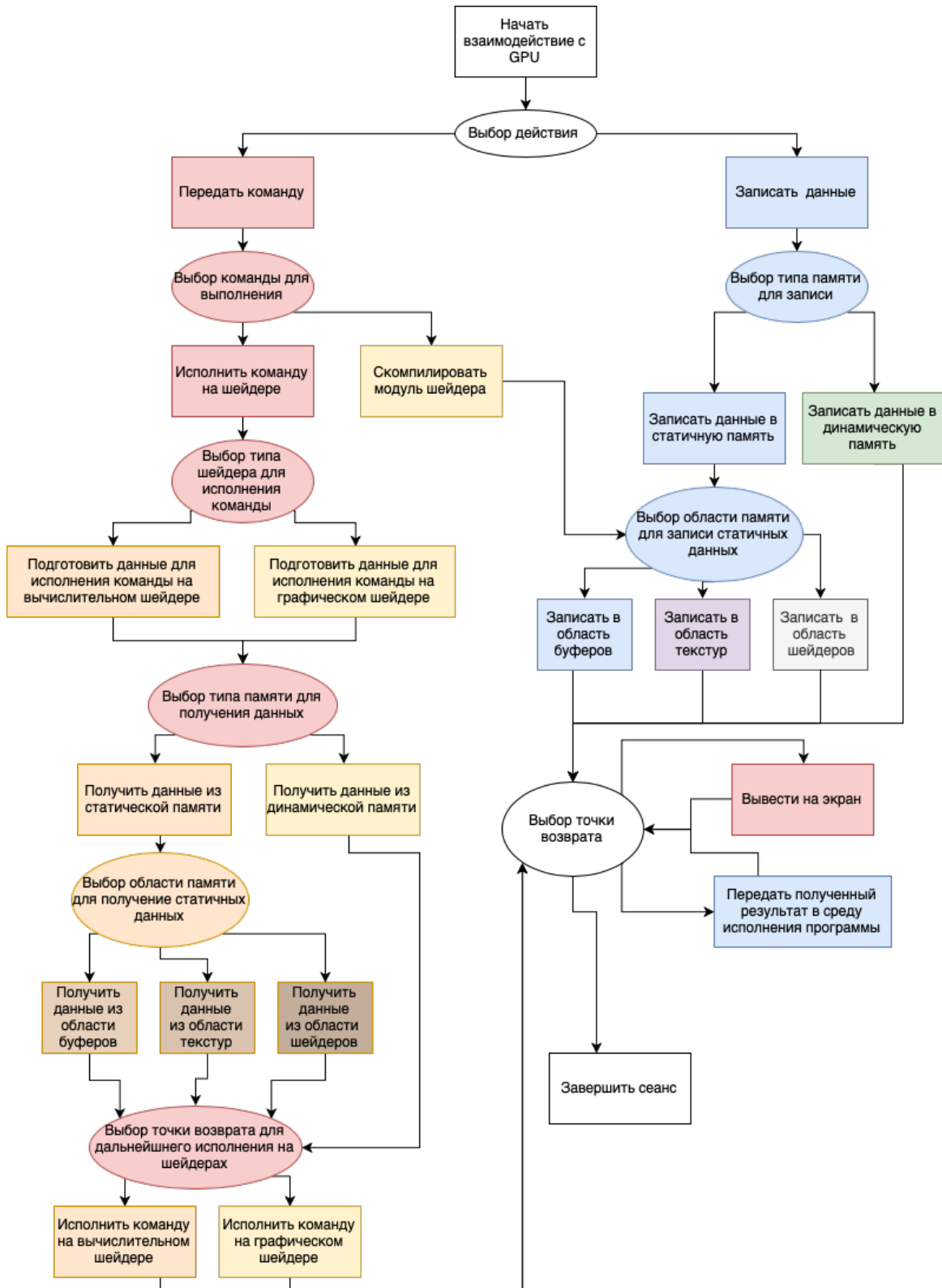


Рис. 4.4. WF-сеть (полный вариант)

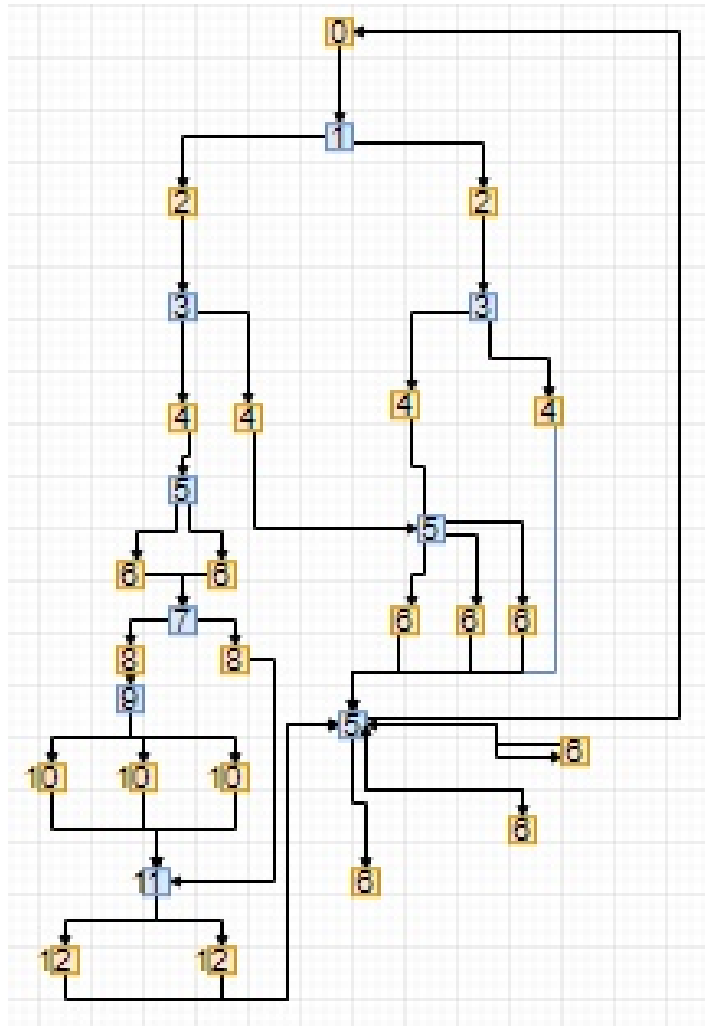


Рис. 4.5. WF-сеть (упрощенные обозначения)

Таблица 4.1. Вершины первого типа

№	Название операции	Характеристика результата успешного выполнения	Характеристика результата неуспешного выполнения (завершения процесса и перехода к точке выхода)
1	Начать взаимодействие с GPU	Вернуть ID устройства, с которым необходимо взаимодействовать	GPU отсутствует
2	Передать команду	Сообщение о передаче новой команды	Не предполагается
3	Записать данные	Сообщение о начале записи новых данных	Не предполагается

4	Исполнить команду на шейдере	Сообщение о начале исполнения команды на шейдере	Не предполагается
5	Скомпилировать модуль шейдера	Сообщение об успешной компиляции шейдерного модуля	Компиляция шейдерного модуля прошла с ошибкой, возвращается код ошибки
6	Записать данные в статичную память	Сообщение о начале записи в статичную память	Не предполагается
7	Записать данные в динамическую память	Сообщение о начале записи в динамическую память	Не предполагается
8	Подготовить данные для исполнения команды на вычислительном шейдере	Получение списка необходимых данных	Сообщение о невалидности запрашиваемых данных
9	Подготовить данные для исполнения команды на графическом шейдере	Получение списка необходимых данных	Сообщение о невалидности запрашиваемых данных
10	Записать в область буферов	Вернуть ссылку на участок памяти	Сообщение о нехватке памяти
11	Записать в область текстур	Вернуть ссылку на участок памяти	Сообщение о нехватке памяти
12	Записать в область шейдеров	Вернуть ссылку на участок памяти	Сообщение о нехватке памяти
13	Получить данные из статической памяти	Получить ссылку на участок памяти	Запрашиваемые данные не найдены
14	Получить данные из динамической памяти	Получить ссылку на участок памяти	Запрашиваемые данные не найдены
15	Получить данные из области буферов	Получить ссылку на участок памяти	Запрашиваемые данные не найдены

16	Получить данные из области текстур	Получить ссылку на участок памяти	Запрашиваемые данные не найдены
17	Получить данные из области шейдеров	Получить ссылку на участок памяти	Запрашиваемые данные не найдены
18	Исполнить команду на вычислительном шейдере	Сообщение об исполнении команды на вычислительном шейдере	Сообщение об ошибке во время исполнения
19	Исполнить команду на графическом шейдере	Сообщение об исполнении команды на графическом шейдере	Сообщение об ошибке во время исполнения
20	Вывести на экран	ссылка на буфер кадра	не предполагается
21	Передать полученный результат в среду исполнения программы	ссылка на буфер данных	не предполагается
22	Завершить сеанс	сигнал о завершении	не предполагается

Таблица 4.2. Вершины второго типа

№	Название операции	Параметры управления	Результаты принятия решения (условия переходов к другим операциям)
1	Выбор действия	результат выбора действия, согласно команде GPU.	переход к записи данных или передаче команды для выполнения
2	Выбор команды для выполнения	результат выбора действия, согласно команде GPU.	переход к исполнению и компиляции шейдера
3	Выбор типа памяти для записи	результат выбора действия, согласно команде GPU.	переход к записи данных в статическую или динамическую память
4	Выбор типа шейдера для исполнения команды	результат выбора действия, согласно команде GPU.	переход к подготовке исполнения вычислительного или графического шейдера

5	Выбор области памяти для записи статических данных	результат выбора действия, согласно команде GPU.	переход к записи данных в область буферов или текстур или шейдеров
6	Выбор типа памяти для получения данных	результат выбора действия, согласно команде GPU.	переход к получению данных из статической или динамической памяти
7	Выбор области памяти для получение статических данных	результат выбора действия, согласно команде GPU.	переход к получению данных из области буферов или текстур или шейдеров
8	Выбор точки возврата для дальнейшего исполнения на шейдерах	результат выбора действия, согласно команде GPU.	переход к исполнению вычислительного или графического шейдера, или возврат к получению данных из статической или динамической памяти
9	Выбор точки возврата	результат выбора действия, согласно команде GPU.	переход к выводу на экран или передаче результата в среду исполнения команд или к завершению сеанса

Анализ свойств ИС с применением WF-сети по п. 4.1°

Проанализировав WF-сеть, можно убедиться, что свойство бездефектности, или правильной завершаемости, соответствует следующим требованиям:

- конечная позиция o достижима при любой последовательности переходов от начальной позиции i ;
- WF-сеть не содержит лишних позиций (которые никогда не будут выполнены);
- при достижении конечной позиции i данной сети не должно оставаться меток в промежуточных позициях.

Анализ свойств ИС с применением WF-сети по п. 4.2°

WF-сети также используются для выявления ошибок абстрактного сценария ИС. Был составлен следующий пользовательский сценарий.

Пользователь хочет произвести вычисления на видеокарту (будем считать, что необходимый шейдер для этого уже загружен и скомпилирован). Для этого необходимо:

1. Загрузить данные в динамическую область памяти;
2. Послать команду для начала вычисления на вычислительном шейдере (при этом не стоит забывать, что необходимо получить необходимый шейдер и необходимые данные из области динамической памяти);
3. Получить данные обратно в окружение приложения.

Реализация всех необходимых шагов должна быть прослежена в анимационном режиме WF-сети. Соответствующий проигрыватель нужно самостоятельно найти в сети Интернет.

На основе построенного сценария были проверены три свойства сети:

- сеть должна быть ограниченной. Позиция сети Петри ограничена (k -ограничена), если существует такое целое число k , что число объектов в этой позиции никогда не превышает k . Число k называют емкостью позиции. Сеть Петри ограничена, если ограничены все ее позиции;
- при работе сети не должны появляться неконечные тупиковые состояния, в которых не активирован ни один переход;
- при работе сети не должно возникать "ловушек" – циклов без выхода (объект может попасть в "ловушку", циклически циркулировать в ней, но не может выйти из "ловушки").

Так все свойства для построенного сценария выполнены, то можно утверждать, что рассматриваемая ИС корректна.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем состоят различия между сетью Петри и WF-сетью?
2. Как проверить граф на предмет двудольности?
3. В чем состоит свойство бездефектности сети Петри?
4. В чем состоит свойство ограниченности сети Петри?
5. Почему в правильно работающей сети Петри не должно быть неконечных тупиковых состояний?
6. Почему при работе сети не должно возникать "ловушек"?

Лабораторная работа №5. ИЗУЧЕНИЕ ОСОБЕННОСТЕЙ ИНФОРМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ ПРИ РАЗЛИЧНЫХ ПОДХОДАХ К ЕЕ ОПИСАНИЮ

ЦЕЛЬ РАБОТЫ

Изучить особенности построения систем информатизации деятельности организации (предприятия) при различных подходах к ее описанию.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1°. Производственные процессы в организации разделяются на основные, вспомогательные и процессы управления.

Основные процессы описывают собственно создание продукции (от сбора маркетинговой информации до отгрузки готовой продукции).

Вспомогательные процессы обеспечивают выполнение основных процессов. Это, в том числе:

- сервисное обслуживание оборудования,
- обеспечение энергоресурсами и производственной средой,
- информационное обеспечение,
- обеспечение финансовой поддержки,
- управление окружающей средой,
- PR-деятельность и связь с общественностью;
- процесс подготовки, обучения и аттестации персонала;
- процесс управления документацией. Данный процесс устанавливает порядок разработки, утверждения и ведения документации, регламентирующей деятельность организации и ее отдельных подразделений, который обеспечивает эффективную работу всех процессов.

Процессы управления имеют своей целью выработку и принятие управленческого решения. Они могут приниматься относительно всей организации в целом, отдельной функциональной области или отдельных процессов, например: стратегическое управление; маркетинг; финансово-экономическое управление; логистика и организация процессов; менеджмент качества; персонал.

2°. С точки зрения теории систем в процессе управления выделяются:

- субъект управления (управляющая подсистема) – компонент, формирующий управляющее воздействие;
- объект управления (управляемая подсистема) – компонент, реализующий управляющее воздействие.

Для формирования управляющего воздействия должны быть заданы:

- цель (одна или несколько);
- способы, средства и воздействия для их достижения;
- способ контроля факта их достижения.

Под способами достижения цели понимается технология производства, изготовления и выпуска продукции, изделий, услуг и т. п. Под средствами достижения цели понимается материальные, трудовые ресурсы, производственные мощности. Под воздействием в большинстве случаев, характерных для организаций, понимаются информационные воздействия (административные, моральные и т. п.).

Функция управления – это направление или вид управленческой деятельности, характеризующийся обособленным комплексом задач и осуществляемый специальными приемами и способами.

Выделяются пять общих функций управления: планирование, организация, координация, мотивация и контроль.

Планирование – это непрерывный процесс установления и конкретизации целей развития всей организации и ее структурных подразделений, определения средств их достижения, сроков и последовательности реализации, распределения ресурсов. (Заметим, что в функцию планирования в общем случае не входит задание способов достижения цели).

Организация как функция управления нацелена на то, чтобы претворить намеченные планы в жизнь. Организация непосредственно связана с систематической координацией многих задач и, следовательно, формальных взаимоотношений людей, их выполняющих.

Мотивация – это процесс побуждения себя и других к деятельности для достижения личных целей и/или целей организации.

Контроль – это процесс обеспечения достижения целей. Он состоит из установления норм, измерения полученных результатов и проведения необходимых корректирующих мер.

3° В современных условиях выделяются три подхода к управлению предприятием:

- функциональный
- процессный
- адаптивный (case-менеджмент)

Функциональный подход к управлению основан на закреплении выделенных функций за субъектами управления (должностями или подразделениями). Признаки такого подхода:

- способ решения задачи выбирает субъект в каждом конкретном случае индивидуально.
- есть субъект, который отвечает за рассмотрение входной информации без четкого выделения области ответственности.
- если субъект не может сам решить задачу, он может ее декомпозировать, а также делегировать ее часть (или вообще всю) другому субъекту.

Процессный подход к управлению – это управление через бизнес-процессы, т.е. последовательности действий, направленных на получение заданного результата, ценного для организации. Здесь заранее известен набор действий, субъекты и объекты управления каждого действия, показатели эффективности.

При адаптивном подходе (кейс-менеджмент) варианты построения процессов подгружаются из библиотеки паттернов или создаются вручную по мере их появления в реальной практике. Это позволяет строить управление в условиях исключений и неопределенности, сопровождающих бизнес-процессы в реальной корпоративной жизни.

4°. Необходимость адаптивного подхода чаще всего связана с обработкой исключений в типовом бизнес-процессе, или инцидентов.

Согласно принятому в ИТЛ определению, под инцидентом понимается любое событие, не являющееся элементом нормального функционирования сервиса и при этом оказывающее или способное оказать влияние на предоставление сервиса путем его прерывания или снижения качества.

Типичные примеры и основные категории инцидентов:

- Приложения – сервис недоступен; ошибка в приложении, не дающая клиенту нормально работать; исчерпано дисковое пространство.
- Оборудование – сбой системы; внутренний сигнал тревоги; отказ принтера.
- Заявки на обслуживание – заявки на получение дополнительной информации, совета, документации; забытый пароль.

5°. Для формирования управляющего воздействия субъект управления должен располагать соответствующей информацией (см. п. 2°). Для этого в организации предусматривается документационное обеспечение управления (ДОУ).

Документационное обеспечение управления – это деятельность аппарата управления, охватывающая вопросы документирования и организации работы с документами в процессе осуществления им управленческих функций.

Документ (документированная информация) – зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать.

Объектом ДОУ могут являться отдельные документы и их совокупности (вплоть до всей документации организации).

Вопросы движения и учета документов связаны с понятием документооборота, который может быть бумажным и(или) электронным.

Система электронного документооборота (СЭД) - организационно-техническая система, обеспечивающая процесс создания, управления доступом и распространения электронных документов в компьютерных сетях, а также обеспечивающая контроль над потоками документов в организации.

Аналогом СЭД в англоязычной литературе является Enterprise Content Management (ЕСМ). ЕСМ-система официально определяется как стратегическая инфраструктура и техническая архитектура для поддержки единого жизненного цикла неструктурированной информации (контента) различных типов и форматов. Содержательно под ЕСМ подразумеваются технологии, используемые для сбора, управления, накопления, хранения и доставки информации всем пользователям организации. (Таким образом, ЕСМ – более широкое понятие, нежели СЭД).

6° . Система ДОУ должна соответствовать принятому в организации подходу к управлению. Например:

- при функциональном подходе к управлению в организации есть смысл выделить для ЕСМ объект управления «Поручение» (которые порождаются приказами, обращениями граждан, служебными записками, резолюциями и пр.), унифицировано реализовать само «Поручение» и логику работы с ним в ЕСМ системе. В итоге можно будет комплексно охватить всю функциональную систему управления, единообразными сценариями работы для исполнителей, и при этом масштабировать систему с точки зрения наращивания функций, порождающих поручения;
- при процессном (workflow) подходе к управлению в организации нужно описать последовательность действий для реализации бизнес-процесса (например, согласование договора). Действия – это обычно задачи, которые генерируются для субъектов управления (пользователей, ролей, очередей и т.д.). тогда объекты управления для ЕСМ – это сопроводительные документы и структурированная информация;
- при адаптивном подходе к управлению организацией объектом управления для ЕСМ может служить чек-лист – перечень контрольных точек по задачам, которые необходимо выполнить. По мере исполнения процесса чек-лист может корректироваться и дополняться. Тогда основной функционал информационной системы при адаптивном подходе к управлению – обработка статусов контрольных точек (открытие, закрытие, отмена, деактивирование, назначение и контроль сроков исполнения) плюс информационные сообщения (дискуссии и отчеты о предпринимаемых действиях).

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

1. Выбрать структурное подразделение организации (предприятия) для анализа. Согласовать выбор с преподавателем. Подразделение должно отвечать следующим требованиям:

- Имеется доступ к организационно-распорядительной документации и/или эксперт-консультант по его деятельности.

- Деятельность подразделения не является уникальной – подразделения, решающие такие же или близкие задачи, существуют и в других организациях
 - Размер и сложность деятельности подразделения достаточны для нетривиального анализа. Рекомендуется выбор систем, модель состава которых включает не менее 15 элементов, для которых можно выделить хотя бы два уровня иерархии (элементы могут быть сгруппированы в подсистемы).
2. Изучить деятельность подразделения, определить актуально принятый в нем подход (подходы) к управлению (см. пп. 1°- 3°), обосновать свой вывод.
 3. Выбрать не менее двух объектов управления ЕСМ, характерных для подразделения (см. п. 5°). Построить сценарий их обработки ((схему, граф потоков, граф жизненного цикла и т.п.)) в подразделении as is.
 5. Предложить метрики оценки эффективности управления выбранными объектами, которые должна формировать информационная система. Провести анализ инцидентов, имевших место в анализируемом подразделении.
 - выделить инциденты с объектами управления ЕСМ и инциденты (нештатные ситуации) в бизнес-процессе;
 - описать их в текстовой форме, определить частоту появления и степень важности последствий, определить их место в общей структуре работы подразделения.
 - Предложить изменения в схеме управления и (или) в сценарии работы ЕСМ, позволяющие обрабатывать подобные инциденты более эффективно.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

1. В качестве структурного подразделения для анализа выбрано студенческое радио «Мегабайт» — объединенное студенческое СМИ, которое работает на базе Университета ИТМО. В медиагруппу входят радио, видео, освещение университетских мероприятий, интернет-издание, газета и журнал NewTone. «Мегабайт» ведет репортажи с главных событий вуза и города, общается с исследователями из научных лабораторий и рассказывает о технологических новинках.

В организации присутствуют управляющие должности, которые контролируют работу с клиентами, отдают распоряжения сотрудникам подразделений (Аудио, Видео, Графика, Свет, Новости, Разработка ПО), и взаимодействуют с финансовым потоком.

Сотрудники «Мегабайта» в основном занимаются производством контента не только по своему направлению и в своем подразделении. Так, звукооператор на мероприятии может быть еще и видеооператором. Часто работу для подразделений Видео, Графика и Свет выполняет одно и то же

лицо. Единственные подразделения, которые работают только по своему направлению – Новости и Разработка ПО.

Такая схема управления позволяет максимально вариативно использовать ресурсы организации для решения поставленных задач. Зачастую там, где нужно обычно 5–7 человек, справиться могут 2–3.

В дальнейшей части работы будут затрагиваться тесно связанные подразделения: Аудио, Видео, Свет.

2. В выбранном структурном подразделении ярко выражены 3 подхода к управлению – функциональный, адаптивный и проектный.

Проектный подход отражается в том, что делегирование задач происходит четко внутри подразделения.

Функциональный подход заключается в следующем:

1. Организация имеет четкую структуру управленцев и подразделений;
2. Как решать задачу – остается за подразделением;
3. Первоначально в подразделение приходит задача без четкого разделения зон ответственности.

При этом для выбранного подразделения также характерны признаки адаптивного подхода:

1. Сотрудник конкретного подразделения может временно решать задачи другого подразделения
2. В организации существует много кейсов для одной и той же ситуации, которую могут решить разные подразделения различными методами.

3. Рассматриваются два объекта управления (производится через ИСУ ИТМО):

1. Бронирование актового зала/студии звукозаписи для проведения мероприятия
2. Проведение мероприятия



Рис. 5.1. Бронирование актового зала/студии звукозаписи для проведения мероприятия

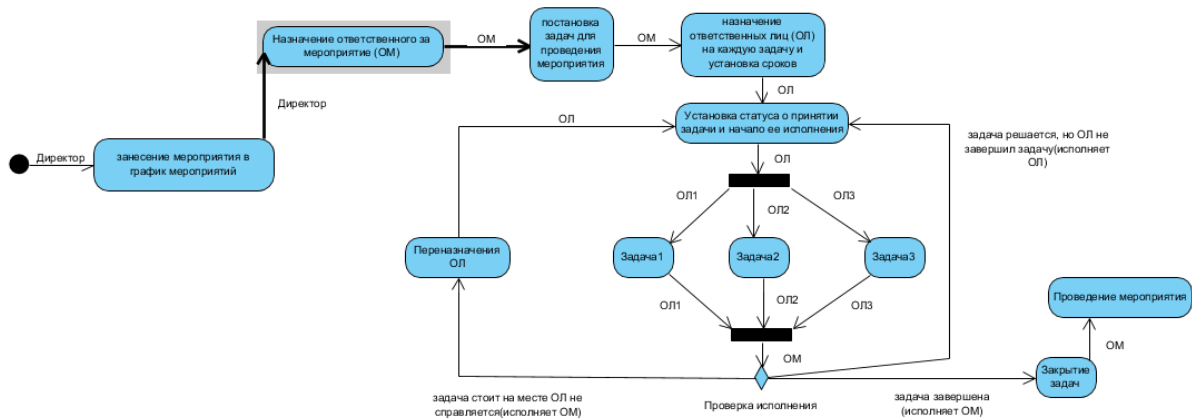


Рис. 5.2. Проведение мероприятия

Бронирование помещения автоматизирует работу с отчетными документами о работе конкретного помещения, закрепленного за Мегабайтом. Также упрощает работу при выстраивания рабочего графика подразделения Аудио, Свет, Видео.

Документальное оформление участия позволяет заполнить мероприятие и его роль в этом мероприятии исходя из занимаемой должности в организации, тем самым студенту не нужно заполнять мероприятия, которые организовывала организация.

Схемы объектов управления представлены на рис. 5.1, 5.2.

4. Для оценки эффективности управления выбранными объектами, которые должна формировать информационная система, предлагаются следующие метрики:

1. самодостаточность, т.е. способность долгое время работать без вмешательства человека;
2. уменьшение количества документов, с которыми приходится работать человеку;
3. упрощение документов до стадии - только подписать (автоматическое заполнение всех полей документов);
4. количество полученных заявок;
5. время отклика на заявку.

5. Проведен анализ возможных инцидентов с объектами управления ЕМС и инцидентов (нештатных ситуаций) в бизнес-процессах.

5.1. Сотрудник подразделения уволился, но участвовал в мероприятии, при этом он остается студентом.

Инцидент влечет за собой проблемы со стороны профкома в случае недобавления студента в список.

Степень важности – высокая, вероятность наступления – средняя.

Один из методов решения – формировать отдельный пул уволившихся сотрудников, который очищать после заполнения документов за прошедший семестр.

5.2. Указанная дата для аудиторией занята и утверждена для важного мероприятия, но Университету необходима она для проведения

собственного мероприятия (Университет - владелец и имеет приоритет по выбору аудиторий в любой ситуации)

Степень важности – высокая, вероятность наступления – очень низкая.

В качестве решения рекомендуется использовать пул резервных скрытых от записи аудиторий, в которые бы можно было перенести мероприятие запланированное раньше. Но обычно Университет заполняет расписание раньше всех, поэтому таких проблем не возникает.

5.3. Заполнение заявки для одного мероприятия на много дат (многодневное мероприятие). В существующем бизнес-процессе нужно заполнять для каждого дня отдельно, есть возможность копирования, но хочется указывать данные в одном месте без лишних манипуляций.

Степень важности – низкая, вероятность наступления – высокая.

Решение – внедрение необходимой функциональности в форму заявки.

5.4. Количество заявок в студию звукозаписи превосходит норму отработки сотрудников.

Степень важности – высокая, вероятность наступления – средняя.

Решение – ограничить время заявки временем работы сотрудника или назначать еще одного сотрудника, который бы мог сменить коллегу после истечения нормы отработки.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие типы производственных процессов существуют в организации?
2. Что должно быть задано для формирования управляющего воздействия?
3. Перечислите общие функции управления.
4. Какие подходы к управлению предприятием выделяются в современных условиях?
5. С чем связана необходимость адаптивного подхода в управлении типовым бизнес-процессом?
6. Дайте определение инцидента в бизнес-процессе.
7. Что нужно описать при процессном (workflow) подходе к управлению в организации?
8. Что может служить объектом управления при адаптивном подходе к управлению организацией ?

Лабораторная работа №6. ОЦЕНКА ХАРАКТЕРИСТИК БИЗНЕС-ПРОЦЕССА С ПРИМЕНЕНИЕМ АППАРАТА СЕТЕЙ ПЕТРИ

ЦЕЛЬ РАБОТЫ

Ознакомиться с возможностями оценки характеристик бизнес-процесса (и поддерживающего его документооборота), предоставляемыми аппаратом сетей Петри.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1° . При использовании аппарата сетей Петри моделируемые процессы описываются множеством событий (действий) и условий, определяющих возможность наступления этих событий, а также причинно-следственными отношениями, устанавливаемыми на множестве пар "события-условия". В сетях Петри условия – это позиции, а события – переходы. Как правило, позиции отражают выделяемые состояния процесса или его компонентов. Наличие метки в одной из позиций соответствует состоянию выполнения некоторой из технологических операций либо состоянию, в котором пребывают некоторые из компонентов процесса.

Переходы соответствуют событиям, отображающим начало или завершение текущего состояния (например, завершение операции транспортирования заготовки к станку и начало операции ее обработки).

Присвоение меток позициям сети Петри называют маркировкой сети. Динамика сетей Петри связана с механизмом изменения маркировок позиций и соглашениями о правилах срабатывания переходов. Переход срабатывает, если в каждой его входной позиции (предусловии) число меток не меньше числа дуг, исходящих из позиции в данный переход (рис. 1, а). Такие переходы называют возбужденными, их срабатывание может наступить через любой конечный промежуток времени после возбуждения. В результате срабатывания из всех входных позиций перехода исключается число меток, равное числу дуг, выходящих из соответствующей позиции в переход, а в выходные позиции данного перехода добавляется число меток, равное числу дуг, исходящих из перехода в соответствующую выходную позицию (рис. 6.1, б).

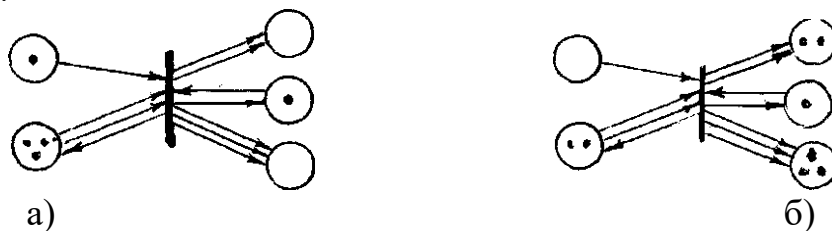


Рис. 6.1. а – возбужденный переход, б – переход после срабатывания

2°. При моделировании производственных процессов позиции отражают отдельные операции (например: транспортировка заготовки к конвейеру, обработка детали на станке, заполнение поля документа) или состояния отдельных компонентов (например, робота, конвейера, станка) процесса. Переходы соответствуют событиям, отображающим начало или завершение моделируемых операций (например, завершение транспортирования заготовки роботом и ее установки на конвейере, подтверждение правильности заполнения документа). Метка, как правило, связывается с сущностью, обрабатываемой в ходе процесса (например, обрабатываемой деталью, сопроводительным документом).

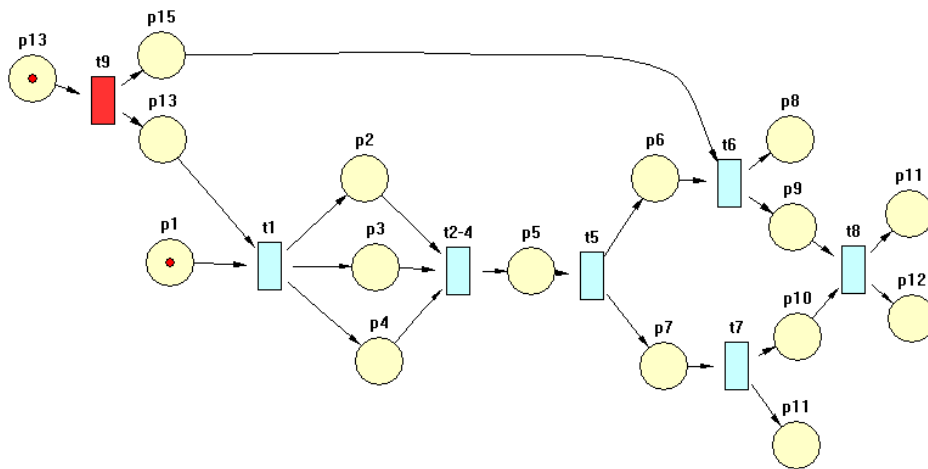


Рис. 6.2. Представление процесса подготовки рукописи в печать в виде сети Петри

В качестве примера на рис. 6.2 показана сеть Петри, моделирующая один из процессов, реализуемых в типичном университетском издательстве – процесс подготовки рукописи в печать (отметим, что в Университете ИТМО реализуется другой процесс). Соотношение компонентов процесса и сети Петри представлено в табл. 6.1

Таблица 6.1. Соотношение компонентов процесса и сети Петри

Компонент сети Петри	Компонент процесса
p1	подготовка и комплектование рукописи
t1	входной контроль пройден
p2, p3, p4	обработка компонентов рукописи (графика, текст, таблицы и формулы)
t2-4	контроль компонентов рукописи пройден (художественный редактор, литературный редактор, технический редактор)
p5	верстка оригинал-макета
t5	подписание оригинал-макета (ответственный редактор)

p6	согласование с учебным отделом
t6	контроль учебного отдела пройден
p7	подготовка электронной копии оригинал-макета
t7	электронная копия оригинал-макета сдана в информационный отдел
p8	формирование отчета о выполнении плана изданий
p9, 10	визирование подписного листа
p11	публикация на портале организации
t8	подписание оригинал-макета в печать ответственным лицом
p11	передача оригинал-макета в типографию
p12	передача подписного листа в издательство

3°. Базовая конфигурация сети Петри позволяет описать:

- возможности параллелизма и синхронизации процессов;
- взаимодействие процессов и ресурсов, представленных метками;
- независимые действия, представленные переходами;

Для учета временных характеристик можно использовать временные сети Петри, в которых каждому переходу t_i сопоставляется время срабатывания τ_i . На множестве переходов в сети Петри может также задаваться отношение порядка, определяющее порядок потребления меток возбужденными переходами в условиях конфликта за метку.

4°. Формальный подход к анализу сетей Петри основан на использовании графа достижимых маркировок (диаграммы состояний сети). Статические свойства системы определяет графовая часть сети Петри, а динамические – начальное маркирование и правила возбуждения переходов. Маркирование M называют достижимым из маркирования M_0 , если существует последовательность срабатывающих переходов $\sigma = t_1, \dots, t_n$, переводящая сеть из состояния M_0 в M . Отношение следования маркирований, возбуждаемых действием последовательности σ , имеет вид $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \dots \xrightarrow{t_{(n-1)}} M_n$. Графическое представление всех состояний сети Петри, достижимых из конкретной разметки, называется деревом достижимости. Дерево достижимости для сети Петри с маркировкой, показанной на рис. 6.2, представлено на рис. 6.3.

Анализ сетей Петри заключается в изучении их основных свойств, определения которых даны в лабораторной работе № 4: безопасность, ограниченность, сохранение, активность, достижимость, покрываемость.

Вопрос ограниченности сети Петри решается перебором и проверкой конечного множества всех достижимых маркировок. Безопасность сети Петри является частным случаем ограниченности.

Свойство сохранения проверяется по дереву достижимости вычислением для каждой маркировки суммы меток. Если метки взвешены, то вычисляется взвешенная сумма. Если сумма одинакова для каждой достижимой маркировки, то сеть является сохраняющей.

$M(p_2, p_1, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{11}, p_{12}, p_{13}, p_{13}, p_{15})$

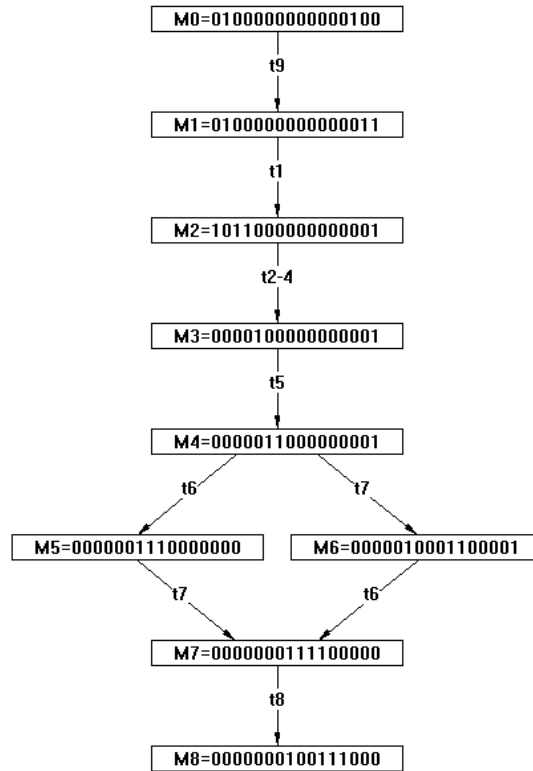


Рис. 6.3. Дерево достижимости для сети рис. 6.2.

Задача покрываемости маркировки M маркировкой M' сводится к поиску на дереве такой вершины x , состояние которой покрывает состояние M . Если такой вершины $M(x)$ не существует, маркировка M не покрывается никакой достижимой маркировкой.

Сеть, дерево достижимости которой содержит терминальную вершину, не активна. Аналогично искомая маркировка M' в задаче достижимости может встретиться в дереве достижимости, что означает ее достижимость. Кроме того, если маркировка не покрывается некоторой вершиной дерева достижимости, то она недостижима.

5°. Важной характеристикой является время перевода сети из состояния M_0 в M , если с каждым переходом связано время τ_i . Для этого на дереве достижимости нужно отыскать соответствующий путь с вектором запусков счёта срабатывания $S = \{s_i\}$, переводящих сеть из состояния M_0 в состояние M . Тогда время перевода сети из состояния M_0 в состояние M определится суммой

$$T = \sum_{j=1}^m \tau_j s_j,$$

где s_j – компонента вектора S для перехода t_j (число срабатываний перехода t_j). При использовании этой формулы нужно учитывать возможный параллелизм в срабатывании переходов.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

1. Для сценариев обработки объектов управления ЕСМ, сформированных в лабораторной работе №5, построить таблицы соответствия (см. табл. 1) и реализовать сети Петри (см. п. 1, 2°).

Для построения сетей самостоятельно найти в Интернете программу, позволяющую не только графически построить сеть Петри, но и запустить ее на исполнение.

2. Запустить построенные сети Петри на исполнение, проверить их адекватность моделируемым сценариям. Запись видео запуска приложить к отчету по ЛР

3. Для построенных сетей Петри построить деревья достижимости.

Произвести их анализ (см. п. 4°), дать его содержательную интерпретацию в терминах моделируемого бизнес-процесса.

4. Провести экспертную оценку времен исполнения отдельных операций анализируемого процесса (см. п. 3°), рассчитать характерные времена исполнения всего процесса (см. п. 5°).

5. Выполнить пп. 1–4 для рассмотренных в лабораторной работе №4 сценариев обработки исключений. Произвести качественное и количественное сравнение показателей по пп. 3, 4.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

В работе рассматриваются два сценария, сформированные в ЛР №5.

Сценарий 1 – бронирование актового зала/студии звукозаписи для проведения мероприятия – представлен на рис. 6.4. Сеть Петри для этого сценария представлена на рис. 6.5, таблица соответствия для нее показана в табл. 6.2.

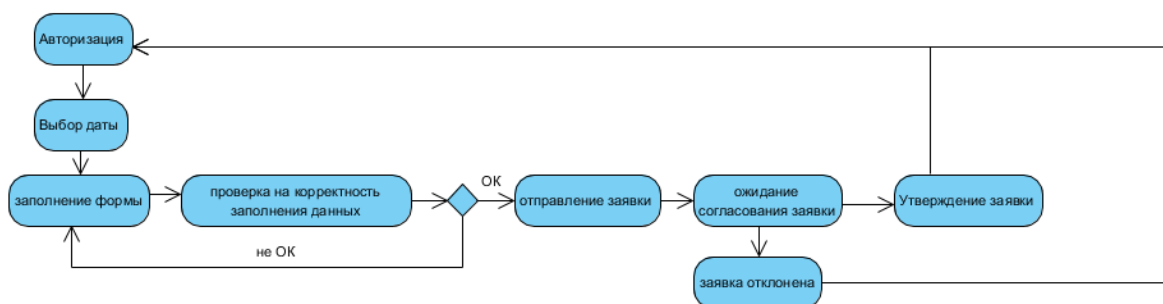


Рис. 6.4. Сценарий обработки бронирования аудитории

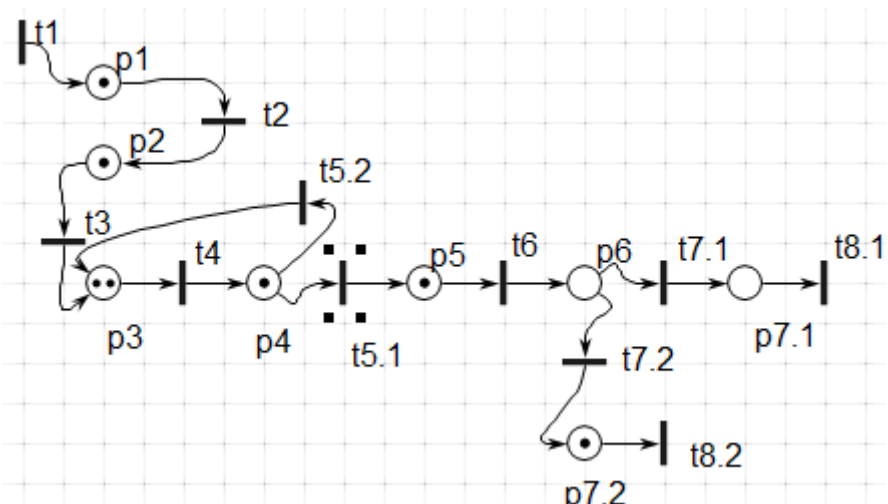


Рис. 6.5. Сеть Петри для бронирования аудитории

Таблица 6.2. Таблица соответствия для бронирования аудитории

Компонент сети Петри	Компонент процесса
t1	
p1	авторизация
t2	авторизация выполнена
p2	выбор даты
t3	дата выбрана
p3	заполнение формы
t4	форма заполнена
p4	проверка на корректность заполнения данных
t5.1	данные корректны
t5.2	данные некорректны
p5	отправление заявки
t6	заявка успешно отправлена
p6	ожидание подтверждения
t7.1	подтверждение получено
t7.2	заявка отклонена
p7.1	утверждение заявки
p7.2	отклонение заявки
t8.1	заявка утверждена
t8.2	

Построенная сеть Петри была запущена на исполнение, проверена ее адекватность моделируемому сценарию. Запись видео приложена к отчету по ЛР.

Построено дерево достижимости для рассматриваемой сети Петри, представленное на рис. 6.6. Анализ дерева достижимости показывает, что сеть функционирует согласно сценарию, показанному на рис. 6.4. На переходе t_5 можно прервать операцию, что может означать отказ клиента от подачи заявки. Выявлены следующие характеристики сети:

1. ограничена;
2. обладает свойством сохранения, так как любой путь содержит одинаковую взвешенную сумму;
3. обладает достижимостью;
4. не активна (имеются терминальные вершины).

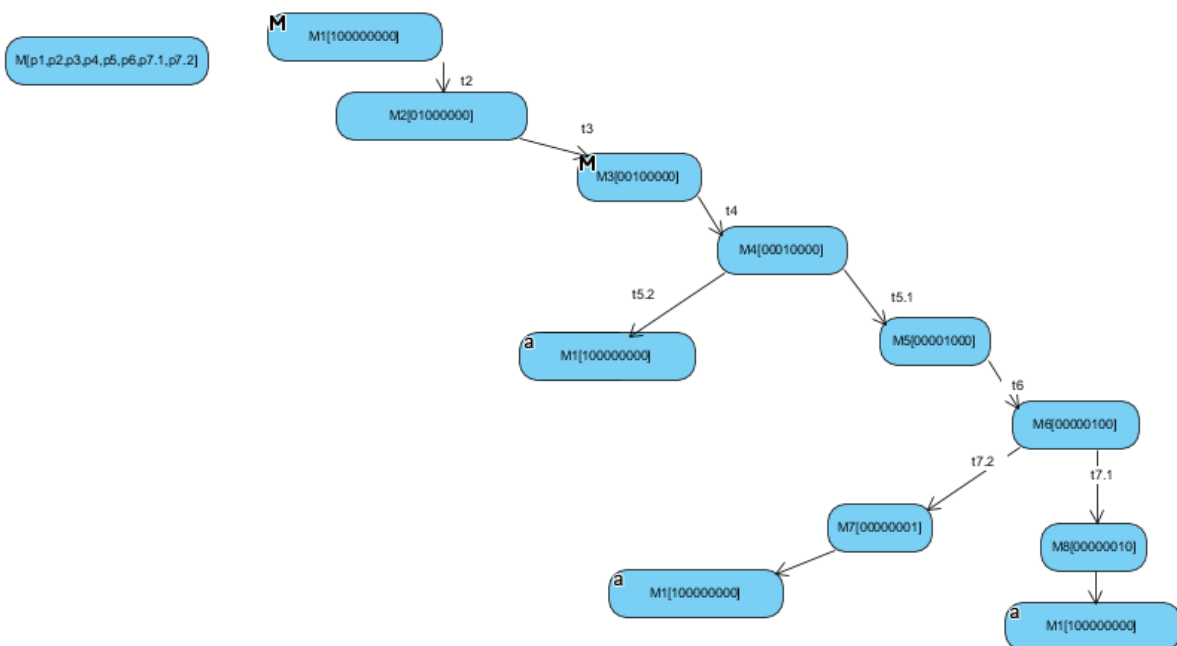


Рис. 6.6. Дерево достижимости для процесса бронирования аудитории

Проведена экспертная оценка времен исполнения отдельных операций анализируемого процесса, рассчитаны характерные времени исполнения всего процесса.

Экспертная оценка времени исполнения отдельных этапов процесса:

$\tau_1 - \tau_6 = 0$, так как эти события зависят только от человека и не занимают времени автоматизированной системы;

$\tau_{7.1} = \tau_{7.2} = 1$ у.е.

Общее время исполнения составляет

$$T = 1 \text{ у.е.}$$

(имеются две различные ветки в конце, которые имеют одинаковый результат).

Сценарий 2 – проведение мероприятия – представлен на рис. 6.7. Сеть Петри для этого сценария представлена на рис. 6.8, таблица соответствия для нее показана в табл. 6.3.

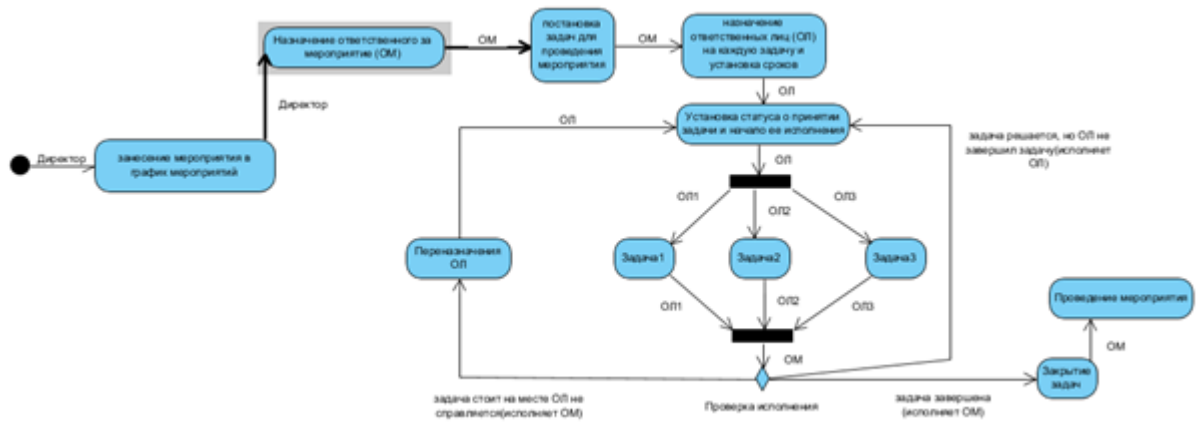


Рис. 6.7. Сценарий проведения мероприятия

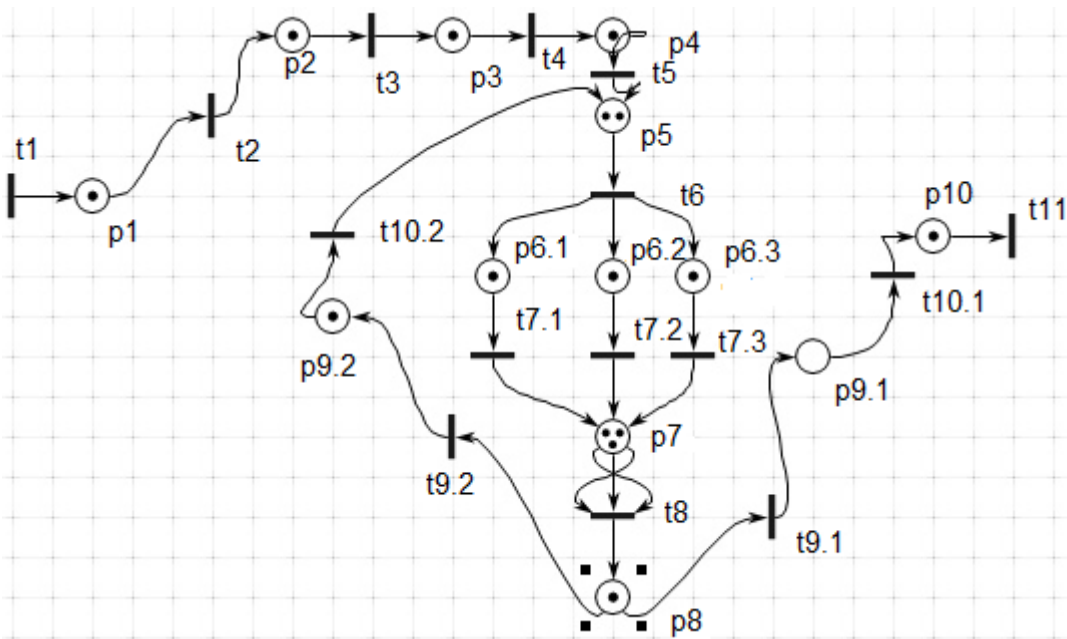


Рис. 6.8. Сеть Петри для проведения мероприятия

Таблица 6.3. Таблица соответствия для проведения мероприятия

Компонент сети Петри	Компонент процесса
t1	
p1	занесение мероприятия в график мероприятий
t2	мероприятие занесено
p2	назначение ответственного за мероприятие (ОМ)

t3	ОМ назначен
p3	постановка задач
t4	задачи поставлены
p4	назначение ответственных лиц (ОЛ) на задачи и установка сроков
t5	ОЛ назначены, сроки установлены
p5	установка статуса о принятии задачи и начало ее выполнения
t6	статус поставлен, выполнение начато
p6.1,p6.2,p6.3	выполнение задачи
t7.1,t7.2,t7.3	дедлайн пройден
p7	составление отчёта о готовности задач
t8	отчет составлен
p8	проверка исполнения
t9.1	мероприятие готово
t9.2	мероприятие не готово
p9.1	закрытие задач
p9.2	корректировка в исполнении задач
t10.1	задачи закрыты
t10.2	корректировка выполнена
p10	проведение мероприятия
t11	мероприятие проведено

Построенная сеть Петри была запущена на исполнение, проверена ее адекватность моделируемому сценарию. Запись видео приложена к отчету по ЛР.

Построено дерево достижимости для рассматриваемой сети Петри, представленное на рис. 6.9.

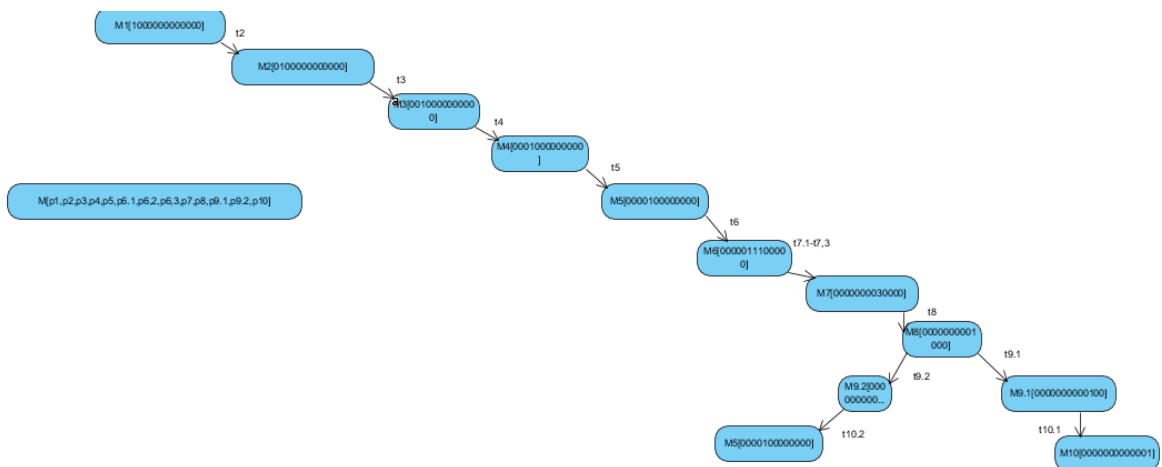


Рис. 6.9. Дерево достижимости для процесса проведения мероприятия

Анализ дерева достижимости показывает, что сеть функционирует согласно сценарию, показанному на рис. 6.7. На переходе t_6 начинается основная работа по исполнению процесса. По истечению текущих дедлайнов проводится митинг, на котором решается, что дальше делать с поставленными задачами. После этого имеет место развилка (это переходы $t_{9.1}$, $t_{9.2}$).

Выявлены следующие характеристики сети:

1. ограничена;
2. обладает свойством сохранения, так как любой путь содержит одинаковую взвешенную сумму;
3. обладает достижимостью;
4. не активна (имеются терминальные вершины).

Проведена экспертная оценка времен исполнения отдельных операций анализируемого процесса, рассчитаны характерные времени исполнения всего процесса.

Экспертная оценка времени исполнения отдельных этапов процесса:

$$\tau_3 = 0.1 \text{ у.е.}$$

$$\tau_4 = 0.3 \text{ у.е.}$$

$$\tau_5 = 0.1 \text{ у.е.}$$

$$\tau_6 = 0$$

$$\tau_{7.1}-\tau_{7.3} = 7 \text{ у.е.}$$

$$\tau_8 = 0.2 \text{ у.е.}$$

$$\tau_{9.1}-\tau_{9.2}, \tau_{10.1}-\tau_{10.2} = 0$$

$$\tau_{11} = 1 \text{ у.е.}$$

Общее время исполнения составляет

$$T = 0.1 + 0.3 + 0.1 + 4 \times (7 + 0.2) + 1 = 30,3 \text{ у.е.}$$

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Поясните содержательный смысл позиций и переходов при моделировании информационных систем сетями Петри.
2. Что такое маркировка сети Петри?
3. Как изменяется конфигурация сети Петри в результате срабатывания перехода?
4. На чем основан формальный подход к анализу сетей Петри?
5. Как проверяется свойство сохранения сетей Петри?

Лабораторная работа №7. ИНТЕГРАЛЬНАЯ ОЦЕНКА КАЧЕСТВА БИЗНЕС-ПРОЦЕССА С ПРИМЕНЕНИЕМ МЕТРИК

ЦЕЛЬ РАБОТЫ

Ознакомиться с методами интегральной оценки качества бизнес-процесса.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1°. KPI (Key Performance Indicator) – это показатель достижения успеха в определенной деятельности или в достижении определенных целей. В управлении бизнес-процессами KPI являются измерителями результативности, эффективности, производительности бизнес-процессов. KPI рабочих процессов – набор метрик, определяемых для оценки эффективности выполнения рабочих процессов в организации. Соответствующие метрики определяются для каждого процесса и декомпозируются на показатели для отдельных сотрудников, ролей, подразделений и организации в целом. Помимо самой метрики, оценивающей эффективность того или иного процесса, определяется ее целевое значение и шкала оценки фактического показателя метрики к ее целевому значению.

К KPI предъявляются следующие требования:

- Адресная принадлежность. Каждый KPI должен закрепляться за конкретным сотрудником или группой из сферы бизнеса (далее – пользователи), несущими ответственность за соответствующие результаты.
- Правильная ориентация. KPI должны быть привязаны к корпоративным стратегическим целям, ключевым бизнес-процессам и проектам развития.
- Достижимость. Утвержденные показатели и нормативы должны быть достижимы. Достижение цели должно быть связано с приложением значительных усилий, но в то же время вероятность ее достижения должна быть не менее 70 – 80%.
- Открытость к действиям. Значения KPI рассчитываются на основе актуальных данных, то есть пользователи должны иметь возможность вмешиваться в процессы, чтобы улучшить результаты работы, пока время еще не упущено.
- Обеспечение прогнозирования. KPI количественно оценивают факторы, влияющие на стоимость бизнеса, то есть они являются показателями, определяющими желательные будущие результаты.

- Ограниченность. КРІ должны фокусировать усилия исполнителей на достижении нескольких высокоприоритетных задач, а не рассеивать их на слишком многие предметы.
- Легкость восприятия. КРІ должны быть легкими для понимания.
- Сбалансированность и взаимосвязанность. КРІ должны быть сбалансированы и «поддерживать» друг друга, а не конфликтовать друг с другом.
- Инициирование изменений. Измерения КРІ должны вызывать в организации цепную реакцию положительных изменений, особенно если за процессом следит руководство компании.
- Простота измерения. КРІ должны быть понятны и доступны для измерения пользователям.
- Подкрепленность соответствующими индивидуальными стимулами. Показатели должны способствовать мотивации пользователей.
- Релевантность. Воздействие КРІ со временем ослабевает, поэтому их следует периодически пересматривать и «освежать».
- Сопоставимость. КРІ должны быть сопоставимыми, чтобы одни и те же показатели можно было сравнить в двух подобных ситуациях. Например, нельзя сравнивать значения такого показателя, как средний чек (КРІ – отношение среднедневной выручки к количеству чеков за день), для магазинов одного формата, но расположенных в областном центре и «в глубинке».

2°. Предложены различные подходы к структурированию совокупности КРІ для бизнес-процессов. Примеры:

списочная структура КРІ

1. КРІ результата – сколько и какой результат произвели;
2. КРІ затрат – сколько ресурсов было затрачено;
3. КРІ функционирования – показатели выполнения бизнес-процессов (позволяет оценить соответствие процесса требуемому алгоритму его выполнения);
4. КРІ производительности – производные показатели, характеризующие соотношение между полученным результатом и временем, затраченным на его получение;
5. КРІ эффективности (показатели эффективности) - это производные показатели, характеризующие соотношение полученного результата к затратам ресурсов.

КРІ как четыре сферы (проекции) сбалансированной системы показателей (balanced scorecard – BSC)

финансы; клиенты; процессы; персонал и развитие.

В свою очередь, КРІ в каждой сфере должны быть конкретизированы, например:

КРІ для процесса

F. Финансы

F1. Доля операционных затрат процесса в обороте организации

- F2. Степень выполнения бюджета процесса
- F3. Стоимость выполнения процесса
- F4. Эффективность процесса
- C. Клиенты и продукты
- C1. Результативность процесса
- C2. Степень удовлетворенности внешнего/внутреннего клиента процесса
- C3. Качество результата процесса
- C4. Своевременность предоставления результата процесса
- P. Бизнес-процессы
- P1. Длительность выполнения процесса
- P2. Доля простоев при выполнении процесса
- P3. Качество выполнения процесса
- P4. Степень организационной фрагментарности процесса
- P5. Степень выполнения плана по улучшению процесса
- R. Персонал и ресурсы
- R1. Доля участников процесса соответствующих требованиям
- R2. Степень удовлетворенности участников процесса
- R3. Степень гибкости (универсальности, взаимозаменяемости) участников процесса
- R4. Степень автоматизации процесса
- R5. Степень информационной фрагментарности процесса

KPI для подразделения

- F. Финансы
- F1. Доля операционных затрат отдела в обороте организации
- F2. Степень выполнения бюджета отдела
- C. Клиенты и продукты
- C1. Результативность работы отдела
- C2. Степень удовлетворенности внешнего/внутреннего клиента отдела
- C3. Индекс SLA (Service Level Agreement) отдела
- P. Бизнес-процессы
- P1. Степень соблюдения регламентов отдела
- P2. Степень выполнения планов работ отдела
- R. Персонал и ресурсы
- R1. Доля сотрудников отдела соответствующих требованиям
- R2. Степень удовлетворенности сотрудников отдела
- R3. Степень гибкости (универсальности, взаимозаменяемости) сотрудников отдела
- R4. Степень автоматизации процессов отдела

3°. Рекомендуемое количество реально назначаемых KPI зависит от уровня конкретного участника бизнес-процесса в организационной иерархии. Более детально:

- 10-12 для генерального директора;
- 5-7 для департамента и его руководителя;

- 3-7 для отдела и его руководителя;
- 3-5 на индивидуальном уровне.

В целом среднее рекомендуемое количество КРІ для одного сотрудника = 5 ± 2 .

4°. Выбранные КРІ для каждого структурного уровня должны быть согласованы, т.е. объединены в единый критерий. Для оценки соотношения важности отдельных КРІ необходимо декомпозировать стратегические приоритеты компании до уровня конкретного подразделения. При взвешивании, то есть при выборе целей с большим весом, необходимо ставить акцент именно на достижение стратегических целей конкретного подразделения и КРІ на ближайший период. Кроме того, полезны следующие рекомендации:

- веса надо расставлять, начиная с более важных КРІ.
- КРІ, соответствующие плохо измеряемым и нерелевантным целям, должны иметь относительно небольшой вес (15–20%).

При построении единого критерия возможны следующие принципы объединения:

(1). свертка отдельных КРІ $q_1(x), \dots, q_i(x), \dots, q_p(x), i=1, \dots, p$, в один "суперкритерий" q_0 путем введения аддитивной функции

$$q_0 = \sum_{i=1}^p \frac{\alpha_i q_i}{s_i},$$

где α_i – весовые коэффициенты, отражающие вклад каждого частного критерия в суперкритерий. Выбор коэффициентов α_i осуществляется с точки зрения вклада целей, соответствующих конкретному КРІ, в стратегические цели бизнеса в целом.

(2). Условная максимизация – выделение основного КРІ $q_1(x)$ и нахождение его условного экстремума при условии, что дополнительные КРІ $q_i(x)$ не превосходят заданных уровней:

$$x^* = \arg \left\{ \max_{x \in X} q_1(x) \mid q_i(x) \leq C_i, i = 2, \dots, p \right\}.$$

(3). Установление относительной важности отдельных КРІ методом парных сравнений с дальнейшим использованием принципа (1).

5°. Ниже приведены примеры построения КРІ в соответствии с п. 2°

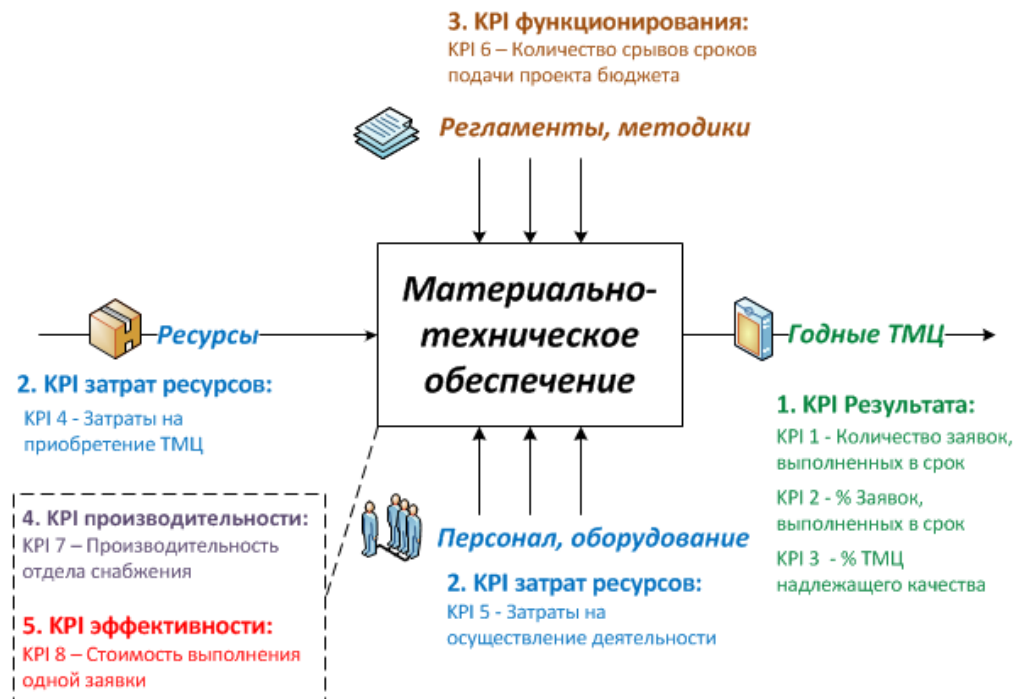
Пример 1. КРІ для отдела рекламы и маркетинга

<u>Перспектива</u> <u>финансы.</u> Учитываются цели,	<u>Подраз-</u> <u>деление</u>	<u>Долж-</u> <u>ность</u> <u>сотруд-</u> <u>ника</u>	<u>КРІ</u>	<u>Формула расчета</u>

связанные с финансовыми результатами(ожиданиями руководства).	Отдел рекламы и маркетинга	Руководитель отдела	Выполнение плана по объему продаж, %	$((SV_{факт.}/SV_{план.}) - 1) \times 100\%$, где $SV_{факт.}$ - объем продаж фактический; $SV_{план.}$ - объем продаж плановый.
			Увеличение маржинальности групп продуктов к аналогичному периоду предыдущего года, %	$((V_{м. п.}/V_{общ.})_{наст. период} - (V_{м. п.}/V_{общ.})_{прош. период}) \times 100\%$, где $V_{м. п.}$ - объем маржинальной продукции; $V_{общ.}$ - общий объем продукции.
Перспектива клиенты. Учитываются цели и КРІ, которые связаны с фактором «удовлетворенность клиентов», в том числе и направленные на устранение недостатков внутренних бизнес-процессов.	Отдел рекламы и маркетинга	Руководитель отдела	Маржинальность марок (Процент маржи), %	$(P_{прод.} - P_{закуп.}/P_{закуп.}) \times 100\%$, где $P_{прод.}$ - стоимость продаж; $P_{закуп.}$ - стоимость закупок продукции.
			Уровень знания торговых марок компании, %	$(N_{зн}/N_{опр}) \times 100\%$, где $N_{зн}$ - количество опрошенных, знающих торговую марку; $N_{опр}$ - общее количество опрошенных потребителей.
			Изменения доли рынка по маркам, %	Из отчета внешних маркетинговых агентств.
			Значимое изменение знания марок, %	$((N_{зн}/N_{опр})_{наст. период} - (N_{зн}/N_{опр})_{прош. период}) \times 100\%$, где $N_{зн}$ - количество опрошенных, знающих торговую марку; $N_{опр}$ - общее количество опрошенных потребителей.
			Количество запусков новых продуктов, шт.	$N_{п}/N_{нп}$, где $N_{п}$ - количество продуктов; $N_{нп}$ - количество новых продуктов.
			Отклонение доли новых продуктов в объеме продаж от плановой, %	$((SV_{нов.}/SV_{общ.})_{факт.} - (SV_{нов.}/SV_{общ.})_{план.}) \times 100\%$, где $SV_{нов.}$ - объем продаж новых продуктов (фактический и плановый); $SV_{общ.}$ - общий объем продаж (фактический и плановый).
		Маркетолог	Доля рынка по маркам, %	Из отчета внешних маркетинговых агентств.
			Оценка дилеров, балл	Анкетирование.
Перспектива процессы.	Отдел рекламы	Руководитель отдела	Отклонение роста продаж групп	$((N_{пп}/N_{общ.})_{наст. период} - (N_{пп}/N_{общ.})_{прош. период}) \times 100\%$, где $N_{пп}$ - объем продаж

Учитываются показатели, характеризующие <i>эффективность</i> <i>ключевых бизнес-процессов</i> компании, подразделения или должности.	и маркетинга	продуктов от планового роста, %	<i>продукции определяемой группы; Нобщ. - общий объем продаж.</i>
		Суммарный вес низкопродаваемых продуктов в общем объеме продаж, %	$(SV_{\text{низкопрод.}}/SV_{\text{общ.}}) \times 100\%$, где $SV_{\text{низкопрод.}}$ - объем продаж низкопродаваемых продуктов; $SV_{\text{общ.}}$ - общий объем продаж.
		Доля брендов в ассортименте производимых и реализуемых продуктов, %	$(N_{\text{бр}}/N_{\text{асс}}) \times 100\%$, где $N_{\text{бр}}$ - доля производимых и реализуемых брендов; $N_{\text{асс}}$ - весь ассортимент производимой и реализуемой продукции.
		Процент наличия фокусного ассортимента в ключевых точках продаж, %	$(N_{\text{налич.}}/N_{\text{план.}}) \times 100\%$, где $N_{\text{налич.}}$ - количество фокусного ассортимента, находящегося в наличии в ключевых точках продаж; $N_{\text{план.}}$ - плановое количество фокусного ассортимента, которое должно быть в ключевых точках продаж.

Пример 2. КРІ для процесса материально-технического снабжения

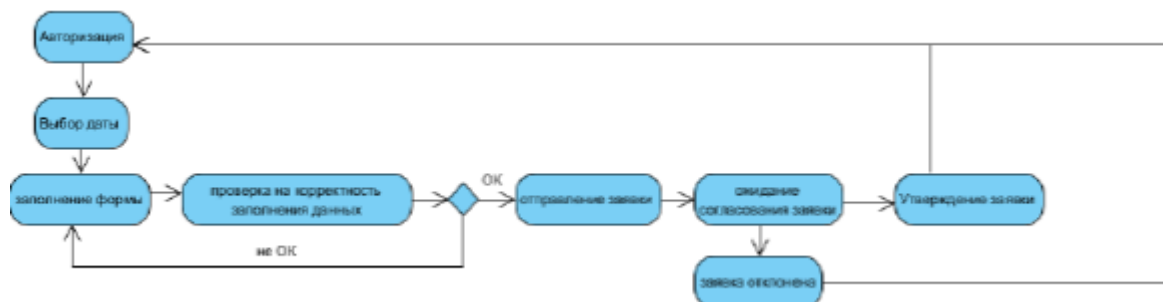


ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ И СОДЕРЖАНИЕ ОТЧЕТА

1. Для сценариев обработки объектов управления ЕСМ, сформированных в лабораторной работе №5, предложить набор КРІ.
2. Проверить соответствие предложенных КРІ требованиям п. 1°.
3. Для каждого из предложенных КРІ составить формулу расчета (см. п. 5°, пример 1).
4. Задать входящие в формулы п. 3 количественные значения коэффициентов, проведя экспертную оценку деятельности подразделения.
4. Построить единый критерий эффективности структурного подразделения (см. п. 4°).
5. Проверить эффективность разработанной системы КРІ в условиях действия исключений, выявленных в лабораторных работах №5, 6.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Сценарий 1. Бронирование актового зала/студии звукозаписи для проведения мероприятия



Для формирования КРІ вводятся следующие обозначения:

- Кобр – количество обработанных заявок;
- Кз – общее количество заявок;
- Зб – количество запросов к системе со сбоем;
- З – общее количество запросов к системе;
- Код – количество одобренных заявок;
- Т – время обработки всех заявок;
- Ккз – количество исправленных заявок.

Предлагаются следующие КРІ.

КРІ результата:

- процент обработанных заявок – $(Кобр / Кз) * 100$ (0.2);
- процент одобренных заявок – $(Код / Кз) * 100$ (0.3);

КРІ функционирования:

- количество сбоев – $Зб / З$ (0.2);

КРІ эффективности:

- среднее время обработки заявки – $Кз / Т$ (0.3).

Сценарий 2. Проведение мероприятия

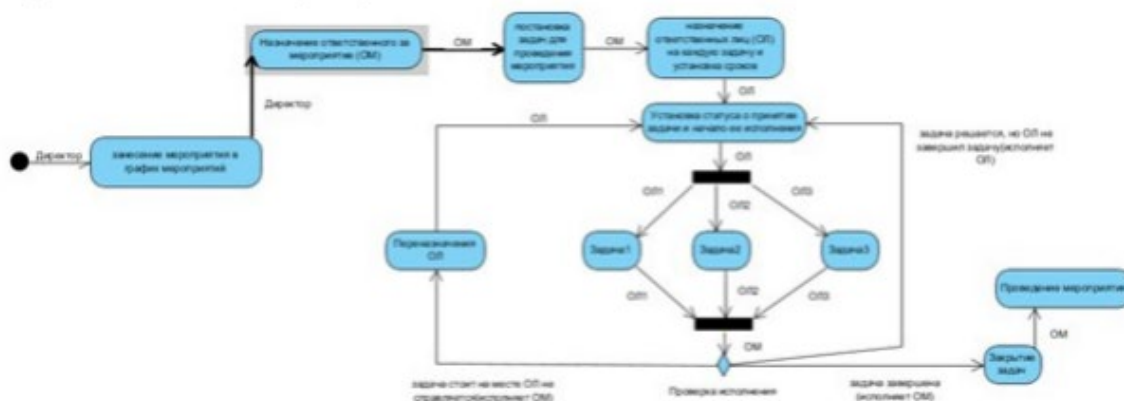


Рис. 2.1. Сценарий проведения мероприятия

Для формирования КРІ вводятся следующие обозначения:

- Сз – затраченная на мероприятие сумма
- Сб – сумма, установленная бюджетом
- Тврзол – сумма времен, затраченных ответственными лицами (ОЛ) на выполнение задач
- Тусз – сумма установленных сроков задач ОЛ
- Квд – количество выполненных дедлайнов
- Код – общее количество дедлайнов
- Кпрм – количество успешно проведенных мероприятий
- Кобм – общее количество мероприятий
- Кнпрм – количество непроведенных мероприятий
- Ксс – количество срывов сроков задач
- Коз – общее количество задач
- Твrm – Сумма времен, затраченных на подготовку мероприятия
- Тус – сумма установленных сроков для мероприятий
- Тврзол – сумма времен, затраченных на выполнение задач ОЛ
- Кзол – количество задач ОЛ
- Кслож – коэффициент сложности задачи
- Кд,*i* – коэффициент, соответствующий пропуску дедлайна у задачи *i*.

Для сотрудников различного уровня (ответственных за мероприятие и ответственных за задачу) предлагаются различные наборы КРІ.

Для ответственного за мероприятие (КРІ ОМ):

- процент срывов сроков выполнения задач – $K_{сс}/K_{оз}$ (0,1);
- нормированное среднее время подготовки мероприятия – $T_{врм}/T_{усм}$ (0,6);
- процент успешно проведенных мероприятий – $K_{прм}/K_{обм}$ (0,1);
- укладывание в бюджет – $S_з/S_б$ (0,2).

Для ответственного лица (КРІ ОЛ):

- уровень сложности задач (назначается ОМ, после обсуждения) –

$$\sum_i K_{\text{слож},i} \times K_{\text{д},i}; (0,3)$$

процент выполненных дедлайнов (определяется по результатам проведения мероприятия) – Квд/Код; (0,6)

- нормированное среднее время выполнения задачи конкретным ОЛ – Тврзол/Тусз (0,1).

При построении единого критерия по обоим сценариям используется свертка отдельных КРІ путем введения аддитивной функции, т.е. суммирования отдельных КРІ со своими весовыми коэффициентами, которые указаны в скобках после каждого КРІ.

Таким образом, заработная плата ОМ и каждого ОЛ складывается из базовой ставки, к которой работник получает надбавку в соответствии с рассчитанным КРІ. Расчёт надбавок и вклада каждого работника происходит после проведения каждого мероприятия.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение КРІ рабочих процессов.
2. Какие требования предъявляются к КРІ?
3. Приведите примеры подходов к структурированию КРІ для бизнес-процессов.
4. От чего зависит рекомендуемое количество реально назначаемых КРІ для конкретного участника бизнес-процесса?
5. Какие цели должны рассматриваться как приоритетные при назначении КРІ на ближайший период?
6. Перечислите принципы объединения частных критериев в единый критерий.

Литература

1. Волкова В.Н., Денисов А.А. Теория систем и системный анализ. Учебник для академического бакалавриата. 2-е изд., пер. и доп. – М.: Юрайт, 2018. – 464 с.
2. Качала В.В. Общая теория систем и системный анализ. Учебник для вузов. – М.: Горячая Линия - Телеком, 2017. – 432 с. – ISBN 978-5-9912-0542-9.
3. Кориков А.М., Павлов С.Н. Теория систем и системный анализ. Учебное пособие. – М.: НИЦ ИНФРА-М, 2019. – 288 с. – ISBN 978-5-16-005770-5
4. Сетевое научное издание «Системный анализ в науке и образовании» (свидетельство о регистрации Эл № ФС77-51141 от 14 сентября 2012 г.).
5. Панов М. М. Оценка деятельности и система управления компанией на основе КРІ. – М.: Инфра-М, 2013. – 255 с. – ISBN 978-5-16-005781-1.

**Гусарова Наталия Федоровна,
Добренко Наталья Викторовна**

ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ

Учебно-методическое пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати 09.10.2019

Заказ № 4343

Тираж 100

Отпечатано на ризографе

Редакционно-издательский отдел

Университета ИТМО

197101, Санкт-Петербург, Кронверкский пр., 49