|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования"МИРЭА - Российский технологический университет"РТУ МИРЭА | |
| Институт информационных технологий (ИТ) | |
| Кафедра промышленной информатики | |

|  |  |
| --- | --- |
| **Отчет по практической работе. Тема:**  **«Итоговый отчет по всем работам»** | |
| **по дисциплине** | |
| **«Проектирование баз данных»** | |
|  | |
| Выполнила студентка группы ИНБО-06-19 |  |
| Принял |  |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | « » 2021 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2021\_ г. |  |

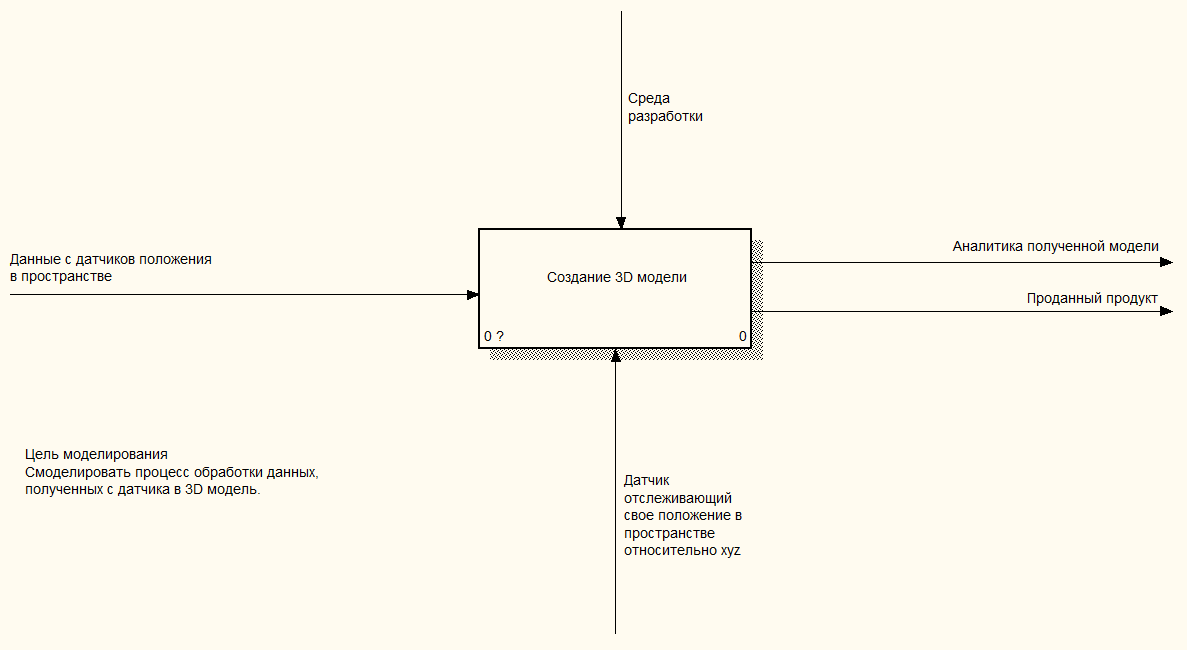
2021 г.

**Задание 1**

*Модель IDEF0 по предметной области «Создание 3D модели».*

**Предметная область «3D модель» —** это объемная, анимированная фигура в пространстве, создаваемая в специальной программе. Анимация представляется в виде последовательности кадров.

На рисунке 1 представлена контекстная диаграмма IDEF0.

Рисунок 1 *IDEF0. Контекстная диаграмма.*

Основной блок – создание 3D модели.

Входной информацией системы является:

* **Данные с датчиков положения в пространстве**

Выходной информацией системы является:

* **Проданный продукт**
* **Аналитика полученной модели**

Механизмы информацией системы является:

* **Датчи отслеживающий свое положение в пространстве относительно xyz;** Управляющие информацией системы является:
* **Среда разработки**

На рисунке 2 представлена декомпозиция контекстной диаграммы.

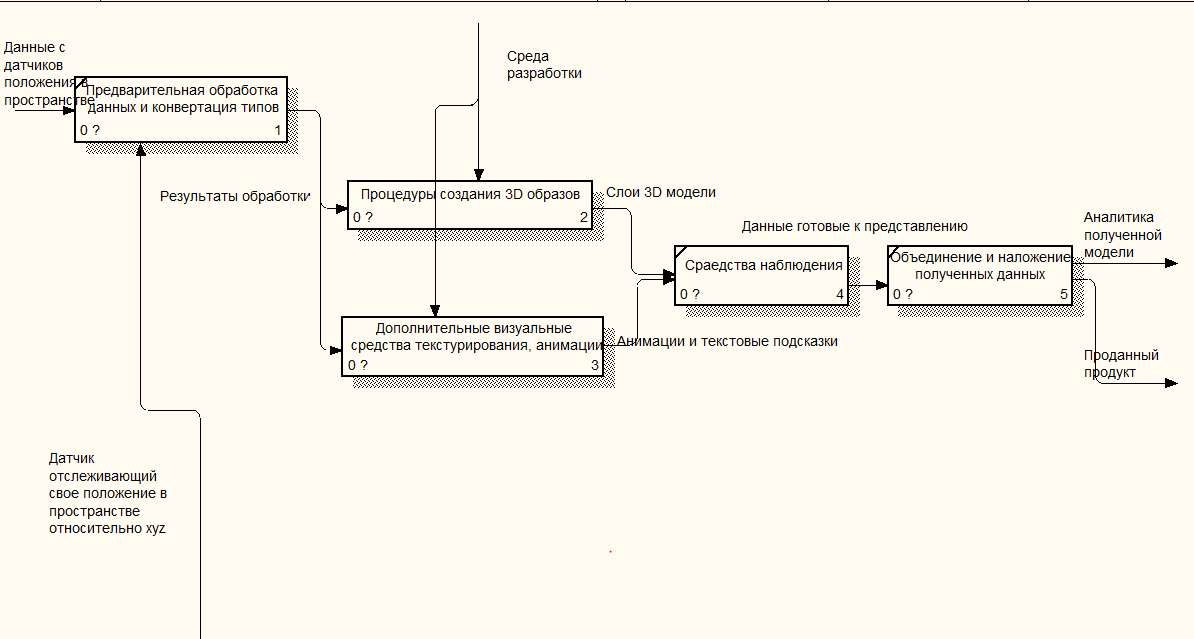


Рисунок 2 *Декомпозиция контекстной диаграммы.*

Мы декомпозировали общий блок «Создание 3D модели» на связанные между собой элементы. В нашем случае делится на 5 основных этапов:

* **Предварительная обработка данных и конвертация типов;**
* **Процедуры создания 3D образов**
* **Дополнительные визуальные средства текстурирования, анимации**
* **Средства наблюдения**
* **Объединения и наложение полученных данных**

На рисунке 3 представлена декомпозиция блока «Процедуры создания 3D образов».

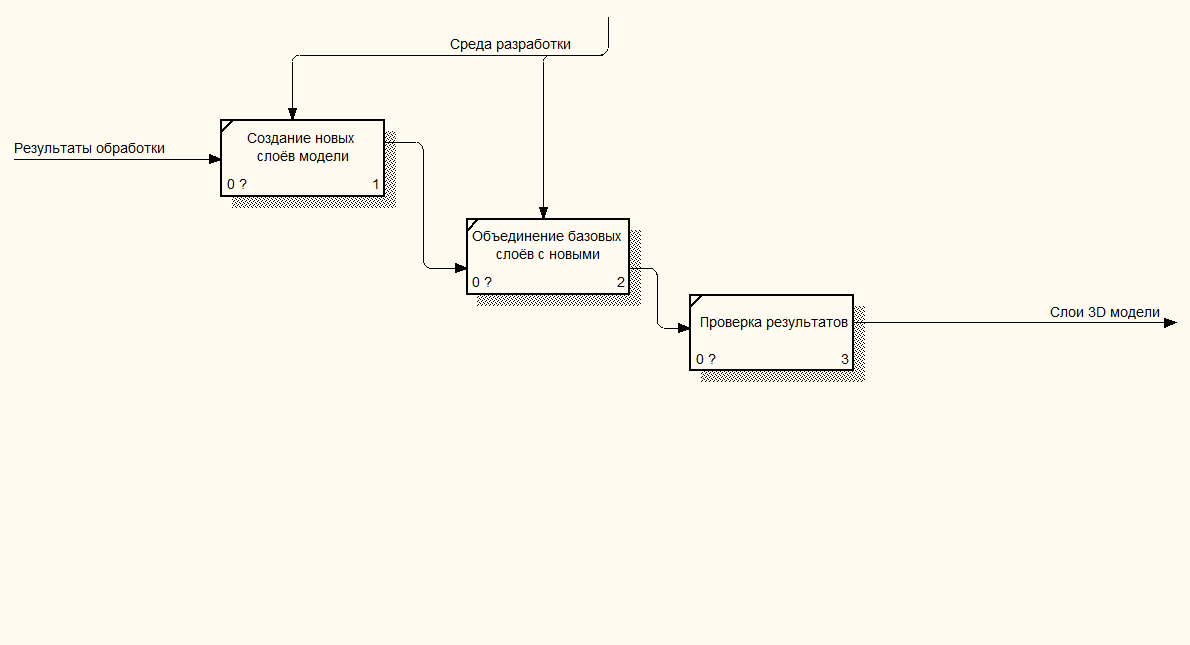


Рисунок 3 Декомпозиция блока «Процедуры создания 3D образов».

Декомпозиция блока «Процедуры создания 3D образов» состоит из трех этапов:

* **Создание новых слоёв модели;**
* **Объединение базовых слоёв с новыми;**
* **Проверка результатов.**

На рисунке 4 представлена декомпозиция блока «Процедуры создания 3D образов».

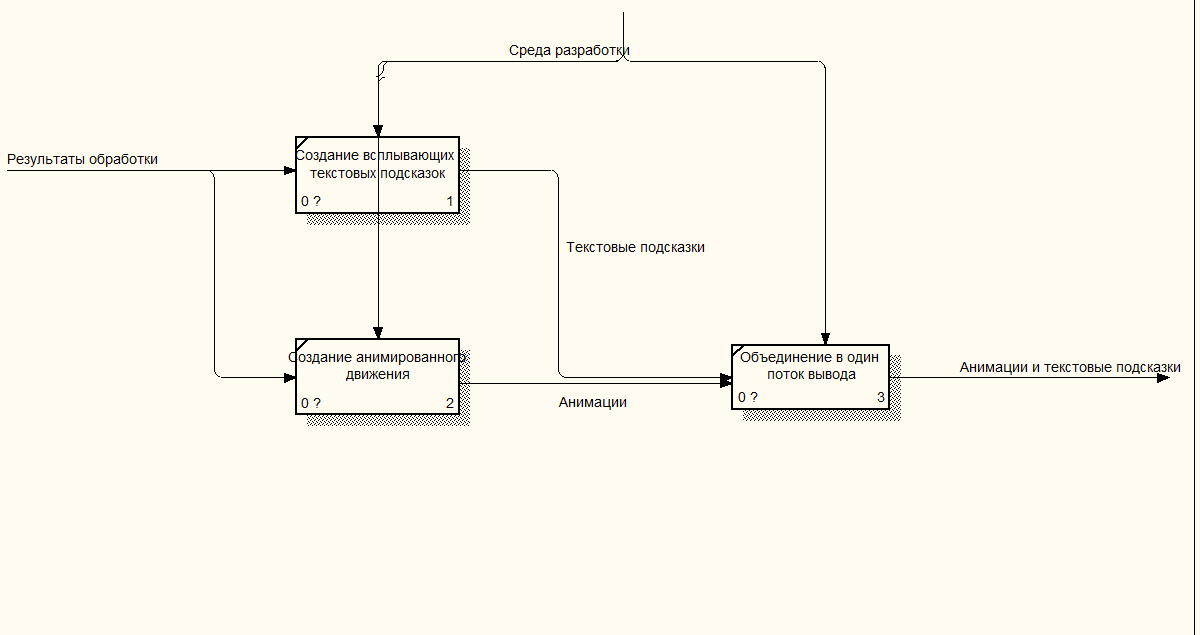


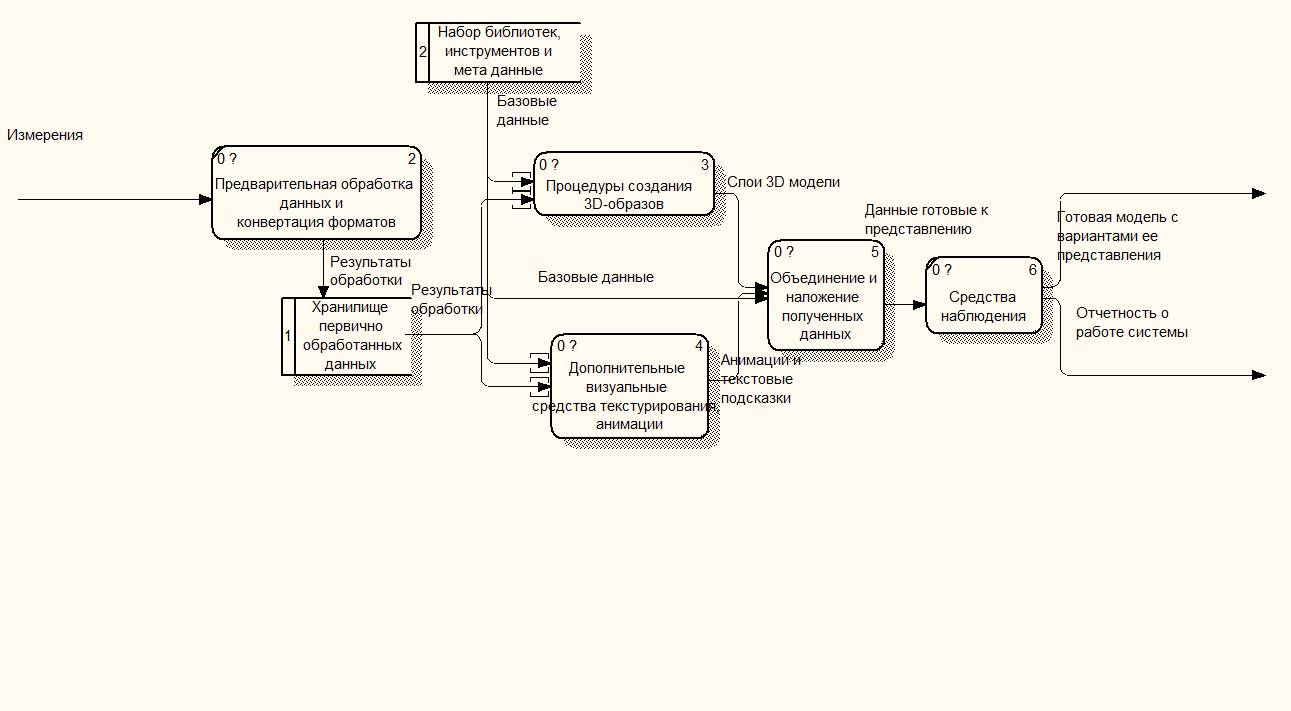
Рисунок 4 Декомпозиция блока «Дополнительные визуальные средства текстурирования, анимации»

Декомпозиция блока «Дополнительные визуальные средства текстурирования, анимации»также состоит из трех этапов:

* **Создание всплывающих текстовых подсказок;**
* **Создание анимированного движения;**
* **Объединение в один поток вывода.**

**Задание 2**

*Модель DFD по предметной области «Создание 3D модели».* На рисунке 5 представлена декомпозиция основного процесса.

 Рисунок 5 декомпощиция основного процесса *DFD*.

На данной модели отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

**Внешние сущности** изображают входы в систему и/или выходы из нее. У нас внешние сущности это : Меню, Отдел закупок, Клиенты, Помещение для кофейни, Сотрудники, Бюджет кофейни.

**Стрелки (потоки данных).** Стрелки описывают движение объектов из одной части системы в другую.

**Хранилище данных.** В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое.

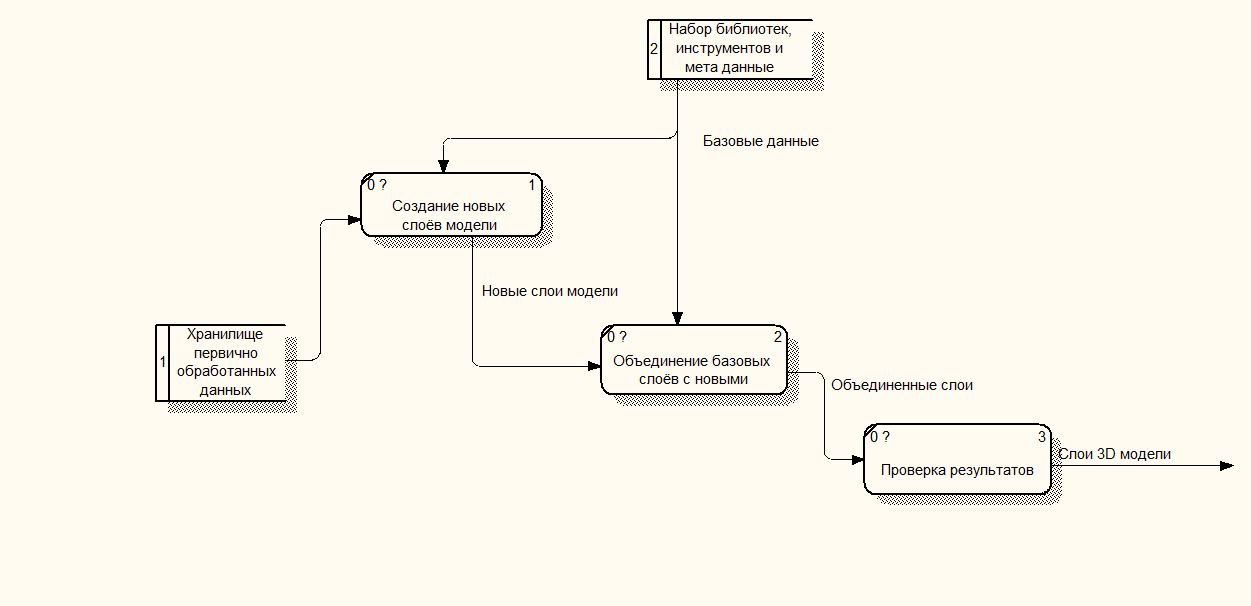


Рисунок 6 декомпозиция блока «Процедуры создания 3D образов».

Блок «Процедуры создания 3D образов» мы декомпозируем еще на 3 этапа:

* **Создание новых слоёв модели;**
* **Объединение базовых слоёв с новыми;**
* **Проверка результатов.**

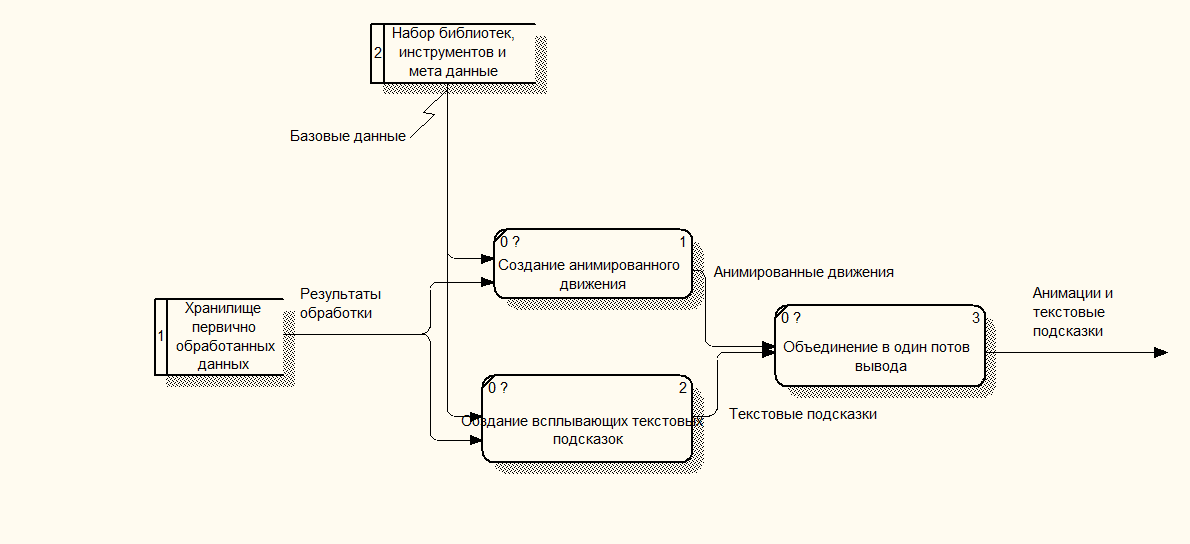


Рисунок 7 декомпозиция блока «Дополнительные визуальные средства текстурирования, анимации».

Блок «Дополнительные визуальные средства текстурирования, анимации» мы декомпозируем еще на 3 этапа:

* **Создание всплывающих текстовых подсказок;**
* **Создание анимированного движения;**
* **Объединение в один поток вывода.**

**Задание 3**

*Модель IDEF3 по предметной области «Создание 3D модели».*

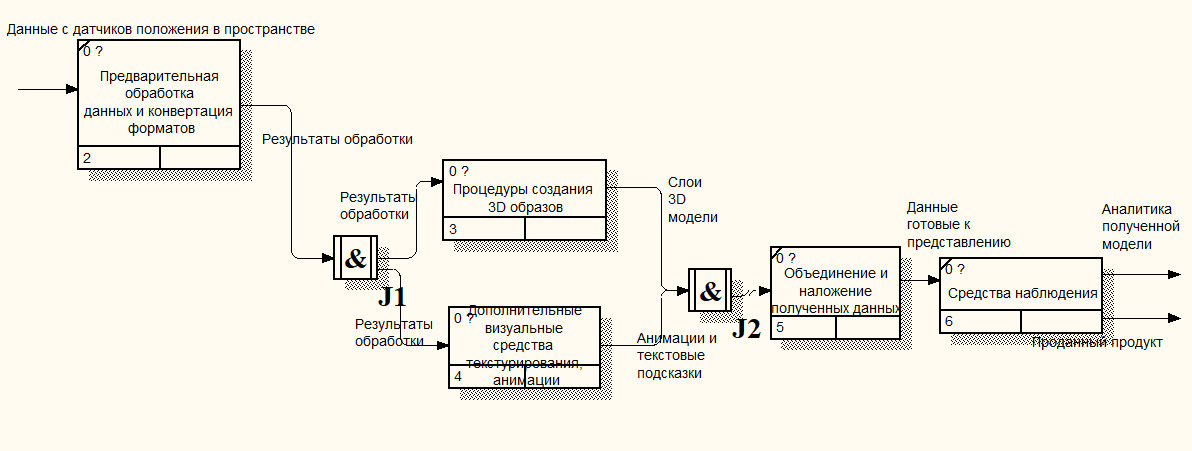


Рисунок 8 декомпозиция предметной области *«Создание 3D модели».*

Методология IDEF3 позволяет декомпозировать работу многократно, т. е. работа может иметь множество дочерних работ. Возможность множественной декомпозиции отражается в нумерации работ: номер работы состоит из номера родительской работы, номера декомпозиции и номера работы на текущей диаграмме.

**Слабые связи переходов** изображаются сплошными одинарными стрелками.

**Сильные связи переходов** изображаются двойными однонаправленными стрелками

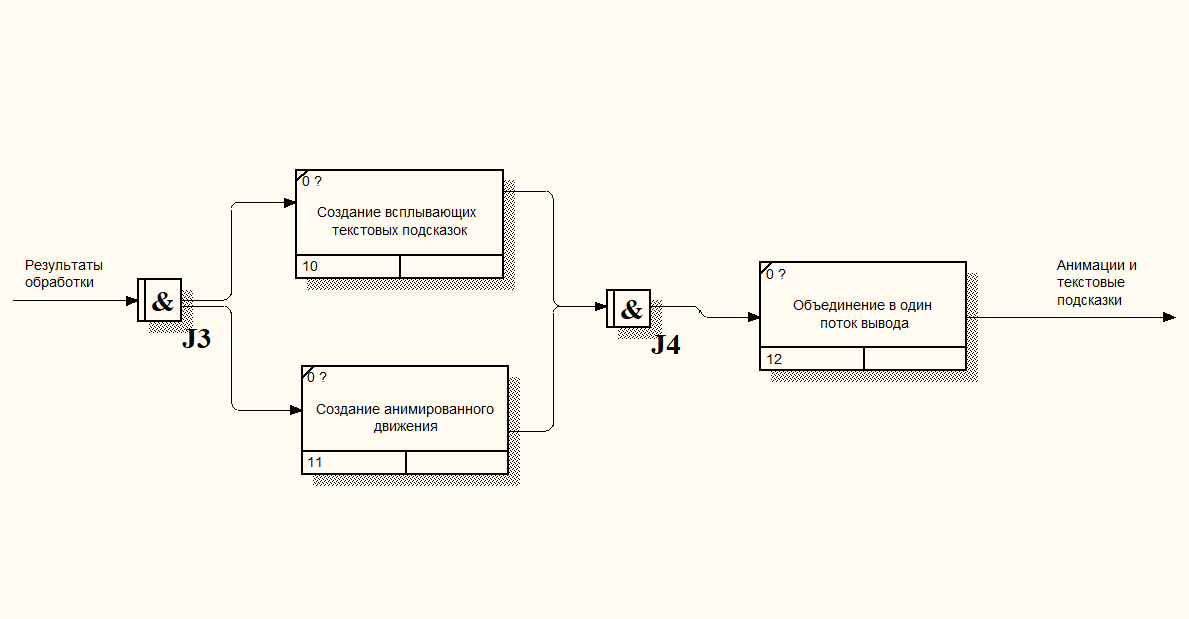


Рисунок 9 декомпозиция блока «Дополнительные визуальные средства текстурирования, анимации»*.*

**Задание 4**

**1. Проектирование информационной системы на языке UML**

Целью данной работы является освоение технологии проектирования информационных систем с позиции объектно-ориентированного проектирования на основе языка UML.

Последовательность работ следующая, строятся диаграммы логического проектирования, не имеющие прямого отношения к языку программирования. Это диаграммы концептуального моделирования.

**1.1 Диаграмма прецедентов**

Use case diagram (диаграммы прецедентов) - этот вид диаграмм позволяет создать список операций, которые выполняет система. Каждая такая диаграмма – это описание сценария поведения, которому следуют действующие лица (Actors).

Данный тип диаграмм используется при описании бизнес процессов предметной области, определении требований к будущей программной системе. Отражает объекты как системы, так и предметной области и задачи, ими выполняемые. Окончательный вид диаграммы показан на рисунке 10. На диаграмме находятся Use Case (Вариант использования), Actor (Действующее лицо). Так же были добавлены ассоциации и описания ко всем элементам диаграммы.

На данной диаграмме мы отобразили организационную структуру вокруг визуализации 3D модели в приложении. Пользователь подготавливает датчик к использованию и в графическом приложении получает информацию с датчика. Разработчик получает отчеты об ошибках и предложения для улучшения системы.

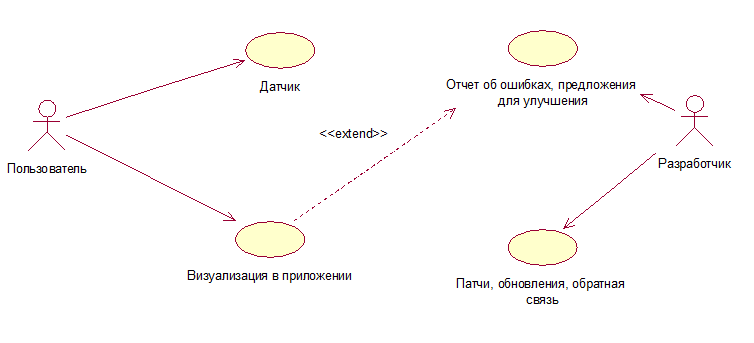


Рисунок 10— Диаграмма прецедентов.

**1.2 Диаграммы п****оследовательности**

Sequence diagram (диаграммы последовательностей действий). Данный тип диаграмм позволяет отразить последовательность передачи сообщений между объектами. Этот тип диаграммы не акцентирует внимание на конкретном взаимодействии, главный акцент уделяется последовательности приема/передачи сообщений. Окончательный вид диаграммы показан на рисунке 11. Диаграмма содержит действующие лица и объекты, так же добавлены такие объекты, как Object Message (Новое сообщение между объектами) и Message to Self (Рефлексивное сообщение самому себе).

На диаграмме показана последовательность действий от подготовки датчика до обновления приложения пользователя.

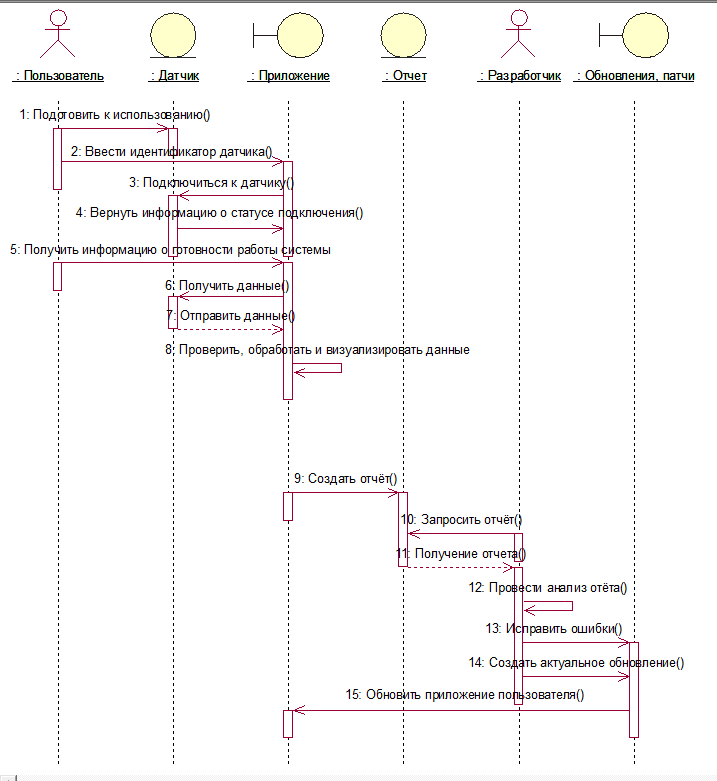


Рисунок 11 — Диаграмма последовательности.

На рисунке 12 отображена последовательность действий пользователя для получения визуализированных данных в приложении.

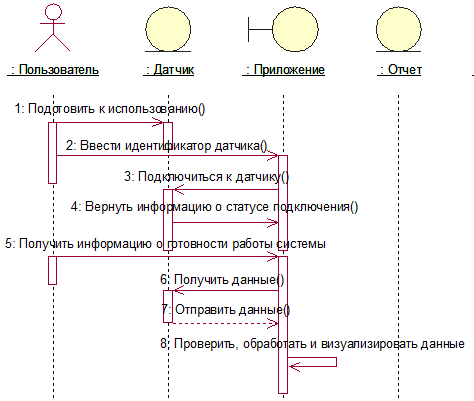


Рисунок 12 — Диаграмма последовательности пользователя для получения данных в приложении**.**

**1.3 Кооперативная диаграмма**

Collaboration diagram (диаграммы сотрудничества). Этот тип диаграмм позволяет описать взаимодействия объектов, абстрагируясь от последовательности передачи сообщений. На этом типе диаграмм в компактном виде отражаются все принимаемые и передаваемые сообщения конкретного объекта и типы этих сообщений. По причине того, что диаграммы Sequence и Collaboration являются разными взглядами на одни и те же процессы, Rational Rose позволяет создавать из Sequence диаграммы диаграмму Collaboration и наоборот, а также производит автоматическую синхронизацию этих диаграмм. Окончательный вид диаграммы показан на рисунке 16. Диаграмма содержит Link To Self (Связь объекта самого с собой), Object Link (Связь между объектами), объекты и Link Message (сообщение, передаваеме между двумя объектами или объектом самому себе).

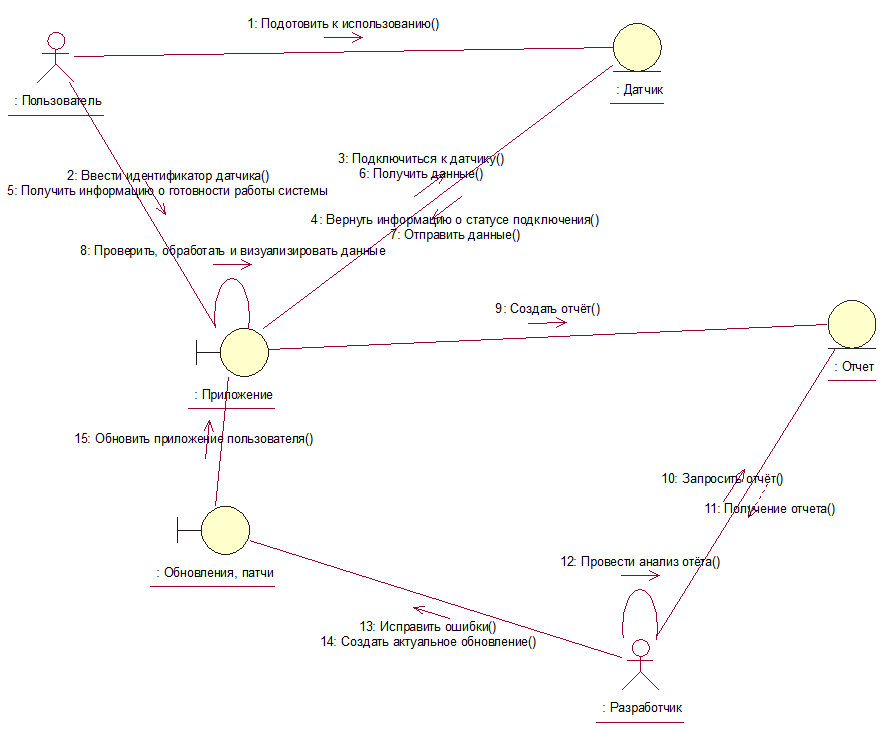


Рисунок 13— Диаграмма коопераций

На рисунке 13 и рисунке 14 изображены диаграммы аналогичные диаграммам последовательности действий.

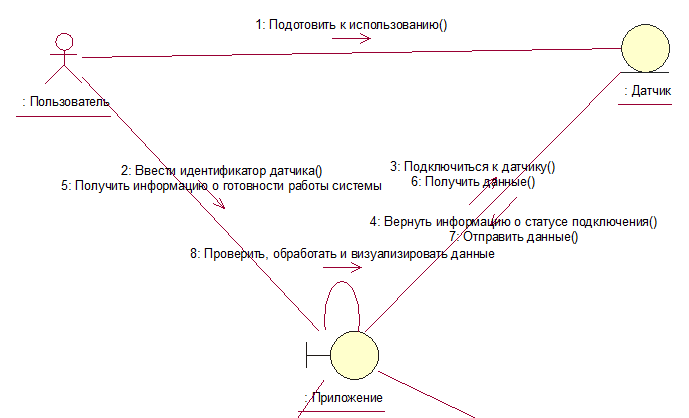


Рисунок 14— Диаграмма коопераций пользователя для получения данных в приложении.

**1.4 Диаграмма Состояний**

Каждый объект системы, обладающий определенным поведением, может находится в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Поведение большинства объектов реальных систем можно представить с точки зрения теории конечных автоматов, то есть поведение объекта отражается в его состояниях, и данный тип диаграмм позволяет отразить это графически. Для этого используется два вида диаграмм: Statechart diagram (диаграмма состояний) и Activity diagram (диаграмма активности) [7].

Диаграмма состояний предназначена для отображения состояний объектов системы, имеющих сложную модель поведения. Конечный вид диаграммы представлен на рисунке 19. Диаграмма содержит в себе state (суперсостояние), состояния, state transition (переходы), начальное (Start state) и конечное состояние (End State).

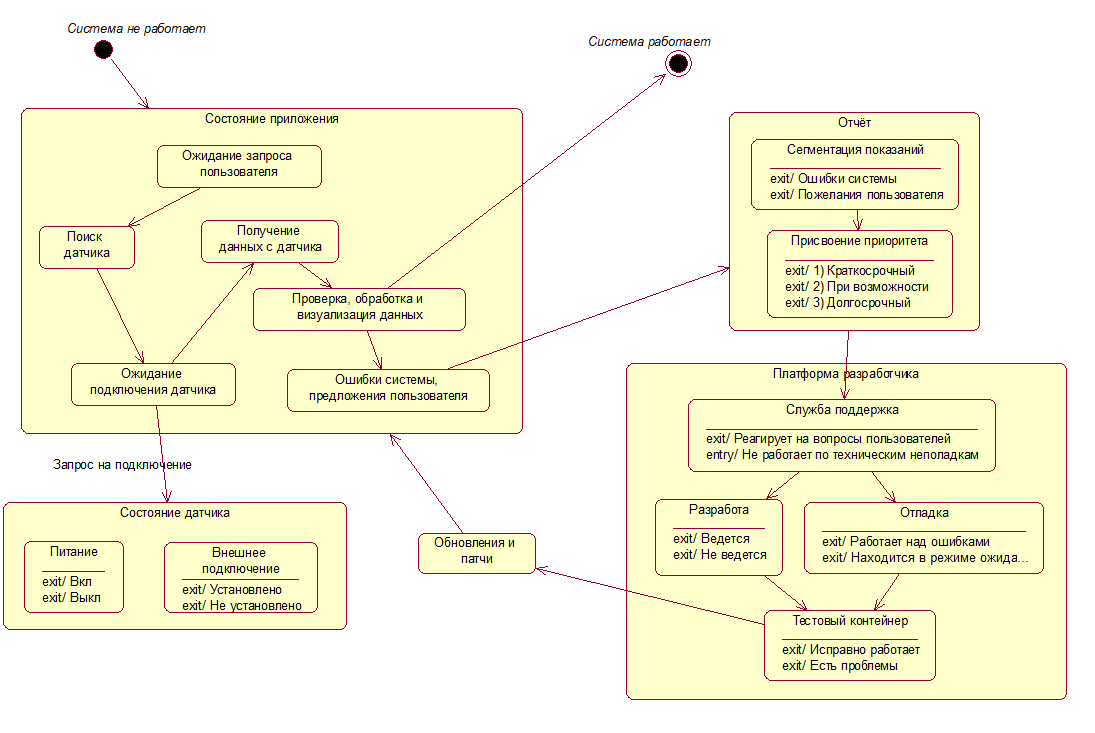


Рисунок 15— Диаграмма состояний.

**1.5 Диаграмма активности для варианта использования «Продажа билетов»**

Activity diagram (диаграммы активности). Это дальнейшее развитие диаграммы состояний. Фактически данный тип диаграмм может использоваться и для отражения состояний моделируемого объекта, однако, основное назначение Activity diagram в том, чтобы отражать бизнес-процессы объекта. Этот тип диаграмм позволяет показать не только последовательность процессов, но и ветвление и даже синхронизацию процессов [7]. Окончательный вид диаграммы показан на рисунке 15. На диаграмме присутствуют дорожки (Swimlane), объекты состояния(Activity), начальное (Start state) и конечное состояние (End State).

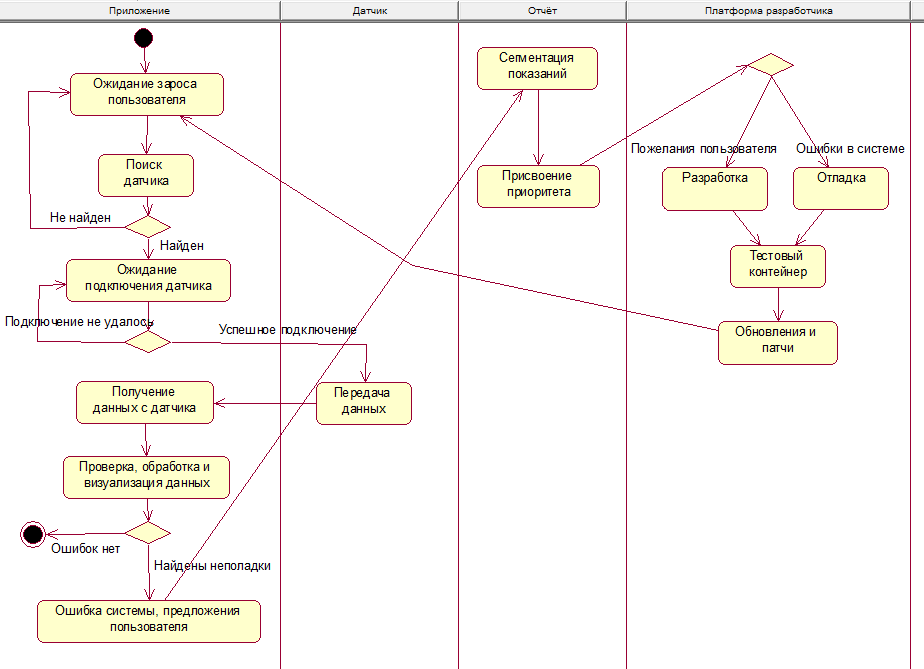


Рисунок 16— Диаграмма активности

**1.6 Пакеты и классы**

Class diagram (диаграммы классов). Этот тип диаграмм позволяет создавать логическое представление системы, на основе которого создается исходный код описанных классов [7].

Значки диаграммы позволяют отображать сложную иерархию систем, взаимосвязи классов (Classes) и интерфейсов (Interfaces). Данный тип диаграмм противоположен по содержанию диаграмме Collaboration, на котором отображаются объекты системы. В данной работе были созданы пакеты, и объедиены в них классы (Рисунок 17).

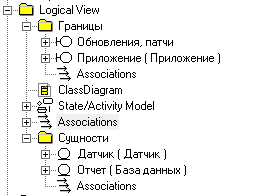


Рисунок 17 — Расположение пакетов и классов в браузере

**1.7 Диаграмма классов**

На данном этапе создается уже готовая диаграмма классов, а именно добавляются новые атрибуты и операции к классу. Происходит подробное описание операций и атрибутов, а именно указываются их типы.

В лабораторной работе 8 происходит описание связей между классами и добавляются ассоциации, далее выбирается множественность связей. В лабораторной работе 9 происходит изменение стереотипов классов. Стереотип позволяет указывать дополнительные особенности для разрабатываемо модели. Понятие стереотипа для этих элементов из уже заданных и представляет собой дополнительную классификацию элементов. В данной работе используется 4 вида стереотипов: Исполнитель (Actor), граница(boundary), сущность(entity) и управление (control). Окончательный вид диаграммы представлен на рисунке 18.

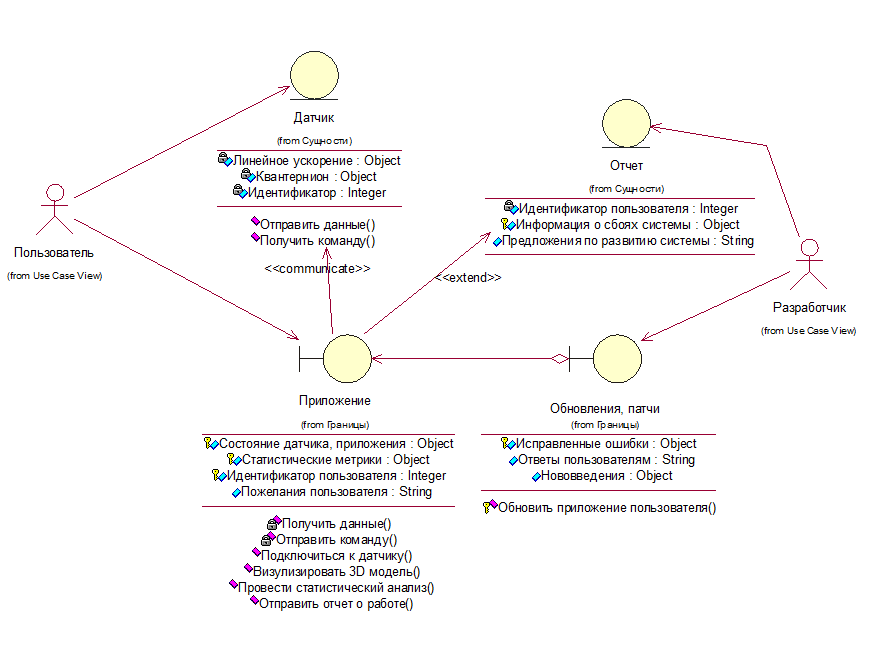


Рисунок 18 — Расположение пакетов и классов в браузере

**1.8 Диаграмма компонентов**

Component diagram (диаграммы компонентов). Этот тип диаграмм предназначен для распределения классов и объектов по компонентам при физическом проектировании системы. Часто данный тип диаграмм называют диаграммами модулей [7]. Диаграмма компонентов представлена на рисунке 19.

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

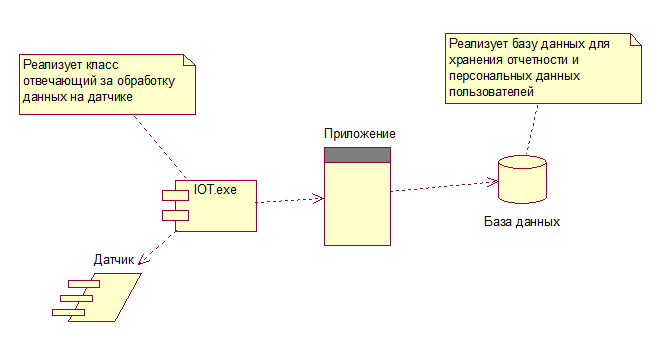


Рисунок 19 — Диаграмма компонент

**Заключение**

В результате выполнения практической работы была достигнута поставленная цель, а именно практическое усвоение основных разделов проектирования информационных систем, развитие практических навыков работы с нотациями IDEF0, DFD, IDEF3 и различными диаграммами в UML, а также изучили правила моделирования потоков данных.

**Список литературы**

1. Боггс У. Боггс М. UML и Rational Rose [Текст]: учеб. Пособие/У.Боггс, М. Боггс:Лори, 2004, - 510с.
2. Кватрани, Т. Rational Rose 2000 и UML [Текст]: учеб. Пособие/ Т. Кватрани: ДМК Пресс, 2001, - 176с.
3. Ларман, К. Применение UML и шаблонов проектирования [Текст]: учеб. Пособие/ К. Ларман: Вильямс, 2002. – 624с.
4. Рамбо Дж., Блаха М.. UML 2.0. Объектно-ориентированное моделирование и разработка. [Текст]: учеб. Пособие/ Дж. Рамбо, М. Блаха: Питер, 2004. – 544с.
5. Трофимов, С. А. - CASE-технологии. Практическая работа в Rational Rose [Текст]/ С. А. Трофимов: Бином-Пресс, 2002. – 288с.
6. Статья С. А. Трофимов UML диаграммы в Rational Rose [Электронный ресурс]. - http://www.caseclub.ru/articles/rose2.html