**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ МОСКОВСКОЙ ОБЛАСТИ**

**«НОГИНСКИЙ КОЛЛЕДЖ»**

**ОТЧЕТ**

**ПО ПМ.05 Проектирование и разработка инфомационных систем**

**УП 05 Учебная практика**

**Тема: «Создание Информационной системы»**

|  |
| --- |
| **Обучающийся группы:**  **Широян Левон Багратович**  **(Фамилия, И.О.)** |
| **Руководитель практики от ОУ**  **Евгений Сергеевич Повираев**  **(Фамилия, И.О.)** |
| **Оценка** |

**2022 г.**

Обучающийся: Широян Левон Багратович гр. 2ИСПР

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc103709603)

[**Описание и сферы применения Ramus Educational** 4](#_Toc103709604)

[**1.1Описание ramus** 4](#_Toc103709605)

[**1.2 Функциональная методика IDEF** 5](#_Toc103709606)

[**1.3 Методология IDEF0. Создание контекстной диаграммы** 8](#_Toc103709607)

[**1.5 Правила построения диаграмм потоков данных — Dfd** 12](#_Toc103709608)

[**1.6 Методика моделирования потоков данных** 17](#_Toc103709609)

[**Глава 2. Процесс создания блок-схем в Ramus** 22](#_Toc103709610)

[**2.1 Методология IDEF0. создание контекстной диаграммы** 22](#_Toc103709611)

[**2.1 Методология IDEF0. создание контекстной диаграммы** 22](#_Toc103709612)

[**2.2 Методология IDEF0. создание декомпозиции контекстной диаграммы** 25](#_Toc103709613)

[**2.3 Методология IDEF0. создание дальнейших диаграмм декомпозиций** 28](#_Toc103709614)

[**2.4 Методология DFD. понятийный аппарат** 33](#_Toc103709615)

[**2.5 Дополнение моделей процессов диаграммой DFD** 35](#_Toc103709616)

[**ЗАКЛЮЧЕНИЕ** 40](#_Toc103709617)

[**БИБЛИОГРАФИЧЕСКИЕ ИСТОЧНИКИ** 41](#_Toc103709618)

[ПРИЛОЖЕНИЕ 1 42](#_Toc103709619)

[Приложение 2 45](#_Toc103709620)

**ВВЕДЕНИЕ**

Программное обеспечение Ramus разрабатывалось для применения в проектах, где требуется выполнить описание бизнес-процессов организации. Ramus способно поддерживать методики моделирования бизнес-процессов IDEF0 (Integration Definition for Function Modeling, то есть, нотация графического моделирования) и DFD (data flow diagrams, то есть, диаграммы потоков данных), а также обладает рядом дополнительных возможностей, которые призваны удовлетворять потребности коллектива проектировщиков систем управления компаниями. Ramus имеет гибкие возможности формирования отчетности по графическим моделям, которые позволяют выполнять отчеты в формате документов, регламентирующих функционирование организации. Ramus Educational обладает достаточно интуитивным интерфейсом пользователя, который позволяет оперативно и просто формировать сложные модели. Ramus Educational обладает меньшим набором функций, чем коммерческая версия программы Ramus, но если требуется сформировать методологию IDEF0, то такая программа может считаться оптимальной для этих целей. Очевидным достоинством программы является тот факт, что она есть в версии на русском языке и её довольно просто использовать.

**Описание и сферы применения Ramus Educational**

* 1. **Описание ramus**

Ramus Educational — программный продукт в области управления знаниями предприятия. Он позволяет выполнять описание, анализ и моделирование бизнес-процессов, а также строить систему классификации и кодирования. Поддерживает стандартные нотации моделирования IDEF0 (функциональное моделирование) и DFD (диаграммы потоков данных). Из преимуществ программы следует отметить: откат/повтор изменений, интуитивно понятный графический интерфейс, связность диаграмм модели по стрелкам, создание отчетов, наличие русского языка интерфейса и модели, отображение туннелирования стрелок. На сайте доступна версия программы для ОС Windows и Linux

Программа Ramus Educational реализует следующие функции:

* Формирование диаграмм стандарта IDEF0.
* Формирование диаграмм DFD.
* Формирование классификаторов
* Осуществление экспорта и импорта в файлы формата IDL.

Процесс моделирования бизнес-процессов организации может быть направлен на разрешение следующих крупных проблем:

1. Подразделение участков ответственности отдельных специалистов и подразделений организации, формирование должностных инструкций и процедур
2. Осуществление анализа, оценки и внесения предложений по усовершенствованию работы организации, реализация методов повышения стандартов качества, в том числе стандартов серии ИСО 9000.
3. Осуществление подготовки и проведения операции сертификации организации на соответствие требованиям международных стандартов качества, в том числе стандартов серии ИСО 9000.
4. Проектирование и внедрение автоматизированной системы управления компанией и ее элементов, а также объединение функционирования различных информационных систем.
5. Формирование и автоматизация внешних взаимодействий, то есть, взаимодействий с клиентами, поставщиками и партнерами.

Для того чтобы решить первые три задачи в основном используются графические модели бизнес-процессов и иногда имитационные модели для реализации более углубленного анализа. Для формирования подобных моделей могут применяться различные графические нотации, такие как, блок- схемы, нотации методологий IDEF0 и IDEF3, ЕРС, диаграммы работы UML, BPMN и другие. Часто используются нотации, которые разработаны в самой организации на базе общепринятых и интуитивно понятных бизнес-пользователям обозначений. Модели данного типа не предполагают высокого уровня строгости и формализации, они прежде всего обязаны быть наглядными, понятными широкому пользовательскому кругу и достаточно информативными.

Формирование моделей, которые служат базой для автоматизации внутренних процессов организации или внешних взаимодействий, предполагает несколько другие требования. Как отмечалось уже выше, передовые концепции управления бизнес-процессами подразумевают автоматизацию непосредственно по модели, которая построена бизнес-аналитиком или менеджером, экспертом в данной предметной области, а не техническими специалистами. Система управления бизнес-процессами должна обладать исполняющим процессором, на который может быть установлено описание бизнес-процесса в форме определённого исполняемого языка и который можно запускать как программу.

**1.2 Функциональная методика IDEF**

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Стандарт IDEF0 (Integration Definition for Function Modeling) утвержден в США в 1993 как Федеральный стандарт обработки информации. В России находится в статусе руководящего документа с 2000 года и в настоящее время в качестве стандарта не утвержден. Тем не менее методология IDEF0 является одним из популярных подходов для описания бизнес-процессов. К ее особенностям можно отнести:

1. использование контекстной диаграммы;
2. поддержка декомпозиции;
3. доминирование;
4. выделение 4 типов стрелок.

Контекстная диаграмма**.**Самая верхняя диаграмма, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется A-0 (А минус нуль). Стрелки на этой диаграмме отображают связи объекта моделирования с [окружающей средой](https://gostehstroy.ru/equipment/vozdeistvie-cheloveka-na-pochvu-kratko-vozdeistviya-cheloveka/). Диаграмма A-0 устанавливает область моделирования и ее границу. Пример диаграммы A-0 приведен на

Поддержка декомпозиции**.**Нотация IDEF0 поддерживает последовательную декомпозицию процесса до требуемого уровня детализации. Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский процесс, но описывает ее более подробно. Согласно методологии IDEF0 при декомпозиции стрелки родительского процесса переносятся на дочернюю диаграмму в виде граничных стрелок.

Доминирование.Блоки модели IDEF0 на неконтекстной диаграмме должны располагаться по диагонали - от левого верхнего угла диаграммы до правого нижнего в порядке присвоенных номеров. Блоки на диаграмме, расположенные вверху слева, "доминируют" над блоками, расположенными внизу справа. "Доминирование" понимается как влияние, которое блок оказывает на другие блоки диаграммы. Расположение блоков на листе диаграммы отражает авторское понимание доминирования. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные.

Выделение 4 типов стрелок**.**Выделяются следующие типы стрелок: "Вход", "Выход", "Механизм", "Управление". Входы преобразуются или расходуются процессом, чтобы создать то, что появится на его выходе. Управления определяют условия, необходимые процессу, чтобы произвести правильный выход. Выходы - данные или материальные объекты, произведенные процессом. Механизмы идентифицируют средства, поддерживающие выполнение процесса. Таким образом, блок IDEF0 показывает преобразование входа в выход с помощью механизмов с учетом управляющих воздействий.

Стрелки обозначают входящие и исходящие из процесса объекты (данные). Каждая сторона функционального блока имеет стандартное значение с точки зрения связи блок-стрелка. В свою очередь, сторона блока, к которой присоединена стрелка, однозначно определяет ее роль. Стрелки, входящие в левую сторону блока - входы. Стрелки, входящие в блок сверху - управления. Стрелки, покидающие процесс справа - выходы, т.е. данные или материальные объекты, произведенные процессом. Стрелки, подключенные к нижней стороне блока, представляют механизмы.

Туннелированные стрелки означают, что данные, передаваемые с помощью этих стрелок, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме. Стрелка, помещенная в туннель там, где она присоединяется к блоку, означает, что данные, выраженные этой стрелкой, не обязательны на следующем уровне декомпозиции. Стрелка, помещаемая в туннель на свободном конце, означает, что выраженные ею данные отсутствуют на родительской диаграмме. Туннелированные стрелки могут быть использованы на диаграммах процессов в нотациях IDEF0, "Процесс", "Процедура".

Элемент обозначает место, сущность или субъект, которые находятся за границами моделируемой системы. Внешние ссылки используются для обозначения источника или приемника стрелки вне модели. На диаграммах внешняя ссылка изображается в виде квадрата, рядом с которым показано наименование Внешней ссылки.  
Внешние ссылки могут быть использованы на диаграммах процессов в любых нотациях.

Элемент обозначает ссылку на типовую модель процесса.  
Наиболее часто повторяющиеся процессы в рамках модели бизнес-процессов могут быть выделены в качестве типовых в отдельную папку в Навигаторе . Диаграмма [типового процесса](https://gostehstroy.ru/business/avtomaticheskoe-regulirovanie-tehnologicheskih-processov-tipovye-shemy/) формируется один раз в одном месте Навигатора. Далее на любой диаграмме может быть использован процесс-ссылка на типовой процесс. Параметры типового процесса заполняются непосредственно в Окне свойств типового процесса. Постоянный список субъектов, принимающих участие в выполнении типового процесса, формируется также в Окне свойств типового процесса.

Список субъектов, принимающих участие при выполнении типового процесса в рамках вышележащего процесса, формируется в Окне свойств процесса-ссылки на типовой процесс.

Процессы-ссылки могут быть использованы на диаграммах процессов в любых нотациях.

**1.3 Методология IDEF0. Создание контекстной диаграммы**

Для того чтобы понять, как работать с функциональным моделированием, я приведу пример процесса написания статьи. Основной блок – «Написать статью».

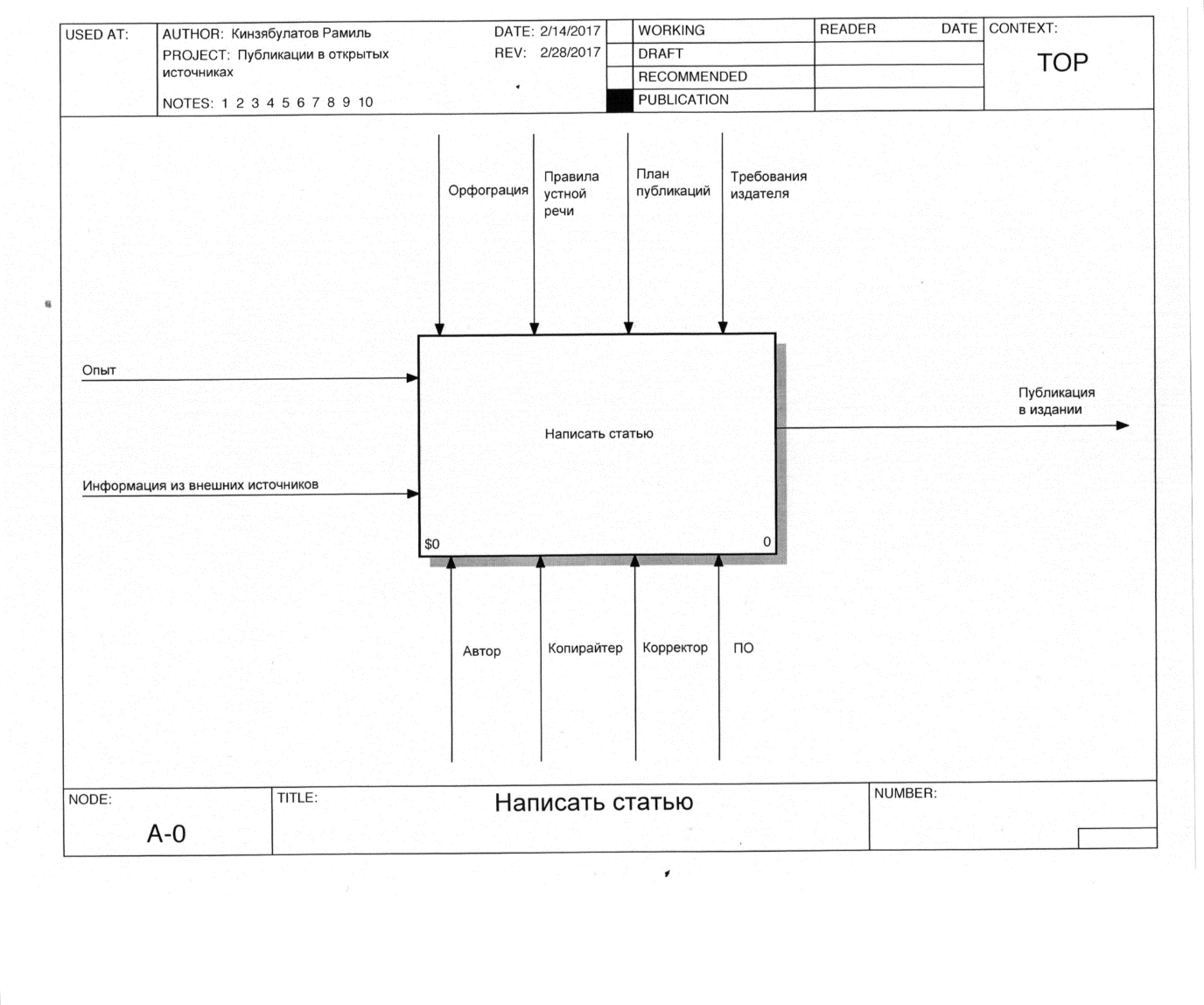


Рисунок 1. Пример создания функциональной модели IDEF0

Входящие стрелки – «Опыт», «Информация из сторонних источников». Это те вводные, которые необходимы для начала работы. Управляющие для написания статьи – это «План публикации», «Требования издателя», «Правила русского языка».

А в роли «Механизмов» выступают автор, копирайтер, корректор и программное обеспечение. В данном случае автор создает аудиоматериал, в котором собирает все мысли и идеи, которые должны быть отражены в статье. Копирайтер – это человек, который создает на основе этого материала, руководствуясь требованиями издателя, планом публикации и правилами русского языка, готовый текст статьи.

Корректор проверяет материал на ошибки. А программное обеспечение – это те инструменты, которые используют в работе все участники процесса. Таким образом, я задал основные параметры процесса, его вход, выход, а также все необходимое для успешного проведения процесса.

Но это – только основные рамки процесса. Так описывается общая схема работы компании в целом. На самом деле, процесс создания статьи, как и любой бизнес-процесс можно и нужно детализировать. Для этого я декомпозирую общий блок «написать статью» на связанные между собой элементы. В нашем случае работа делится на 4 основных этапа:

1. Подготовить аудио.
2. Подготовить текст
3. Подготовить текст к публикации.
4. Разместить статью в издании.

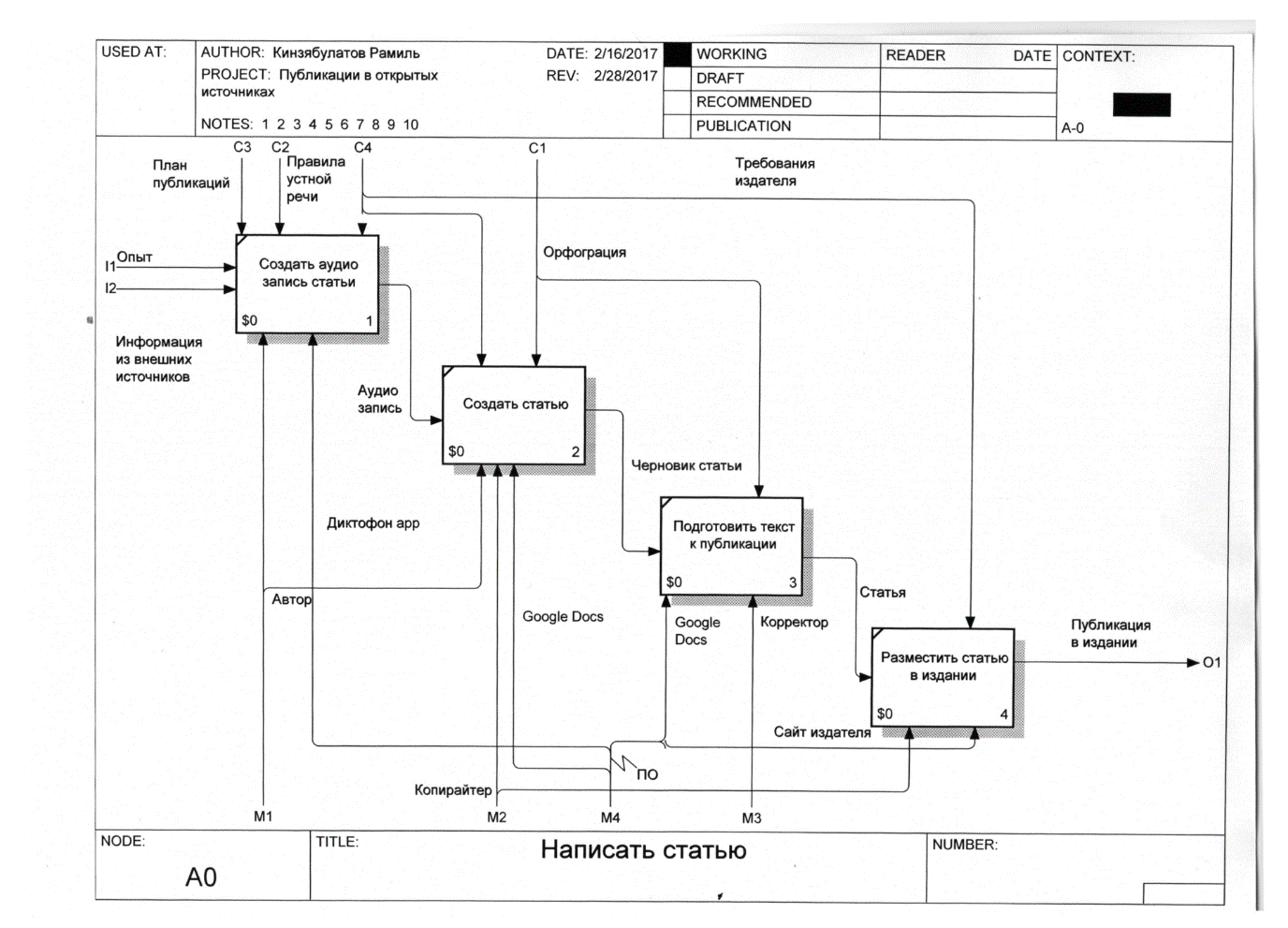


Рисунок 2. Пример создания функциональной модели IDEF0

На схеме наглядно видно, на каком этапе какие управляющие элементы и какие механизмы задействованы.

Так, автор при создании аудио использует свои знания и опыт, при этом руководствуется планом публикации и требованиями издателя. Копирайтер получает на входе аудиозапись, из которой, руководствуясь правилами русского языка, создает текст. Корректор получает текст и проверяет его, также руководствуясь правилами русского языка. Для размещения статьи в издании необходимо специальное программное обеспечение.

При создании функциональной модели ключевыми параметрами являются цель и точка зрения. Исходя из них, моделирование одних и тех же процессов может выглядеть несколько по-разному. Например, в моем случае целью является «рассказать о процессе написания статьи». А точка зрения копирайтера – это «написание и публикация статьи с точки зрения руководителя процесса».

Так, если бы тот же процесс был описан с точки зрения копирайтера, то входящими были бы опыт и аудиофайл от автора. При этом в таком случае под Опытом подразумевался бы опыт копирайтера, но не руководителя или автора. А потому первое, что нужно определить при создании модели бизнес-процесса – это выбрать точку зрения и четко сформулировать цель. Такое моделирование не только наглядно, но и очень удобно для принятия эффективных управленческих решений.

Например, в описанном выше бизнес-процессе есть два отдельных специалиста — копирайтер и корректор. Если я поставлю задачу оптимизировать финансирование проекта, то я благодаря схеме сразу увижу, где это и как можно сделать. Так, к копирайтер и корректор пользуются примерно одинаковыми правилами, но копирайтер получает аудио, а выдает результат в виде текста, корректор же и принимает, и отдает текст. А потому при необходимости я могу, скажем, за половину стоимости обязанности корректора предложить копирайтеру.

Так я сэкономлю средства и время на взаимодействие разных специалистов. Конечно, я понимаю все заслуги корректоров и почему лучше работать с отдельным специалистам. Но напоминаю — у меня стоит задача: оптимизация затрат. Без такого наглядного инструмента было бы сложнее определить, какие из блоков можно удалить и, таким образом, оптимизировать работу.

**1.4 Краткое описание DFD**

Диаграмма DFD наглядно отображает течение информации в пределах процесса или системы. Для изображения входных и выходных данных, точек хранения информации и путей ее передвижения между источниками и пунктами доставки в таких диаграммах применяются стандартные фигуры, такие как прямоугольники и круги, а также стрелки и краткие текстовые метки. Диаграммы DFD варьируются от простейших набросков процессов (включая нарисованные вручную) до подробных многоуровневых схем с глубоким анализом способов обработки данных.

Диаграммы DFD применяются для анализа существующих и моделирования новых систем. В лучших традициях визуализации данных диаграммы DFD часто наглядно «рассказывают» о процессах, которые сложно объяснить словами, и позволяют эффективно донести информацию и до «физиков», и до «лириков», то есть до всех участников организации — от разработчиков до генеральных директоров.

Вот почему диаграммы DFD не утратили популярности за долгие годы существования. Однако стоит упомянуть, что хотя диаграммы DFD отлично подходят для программ и систем потоков данных, в наши дни они далеко не всегда отвечают требованиям ПО и систем, ориентированных на интерактивность, работу в реальном времени и базы данных.

**1.5 Правила построения диаграмм потоков данных — Dfd**

Правила построения диаграмм потоков данных — DFD

DFD – диаграмма потоков данных. Это метод, с помощью которого проводится графический структурный анализ, в котором описаны внешние для системы источники данных, функции, потоки и хранилища данных, к которым имеется доступ. С помощью этой диаграммы проводится структурный анализ и проектируются информационные системы.

Что такое диаграмма потоков данных?

DFD – это нотация, которая используется при моделировании информационных систем с точки зрения хранения, обработки и передачи данных.

В целом DFD не имеет жесткого синтаксиса в отличие от большинства других нотаций. Здесь возможно использование различных вариантов. Главное условие – понятность как составителю диаграммы, так и тем людям, которые будут ее читать.

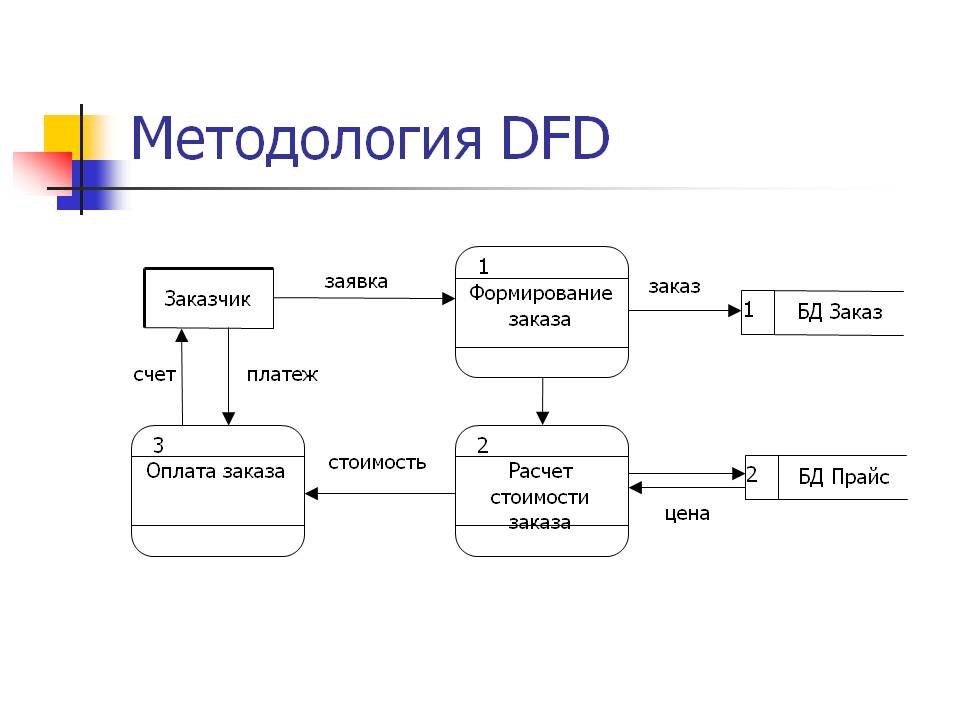


Рисунок 3. Пример создания функциональной модели DFD

Используют нотацию DFD в случаях, когда необходимо описать систему как хранилище данных. Таким образом, целью ее является поиск ответов на вопросы, связанные с составом информационной системы и необходимыми инструментами для обработки информации.

С помощью DFD возможно описание любых действий, например, отгрузку продукции, обработку заявок клиентов с точки зрения описания системы. В результате формируется понимание того, какие элементы должны быть включены в систему, как можно автоматизировать ее процессы. При этом DFD нельзя рассматривать как непосредственно описание процесса. Например, в схему не включают такой важный параметр, как время. Также нет возможности использования условий и развилок.

DFD показывает источник информации, определяет, какие сведения необходимы, порядок обработки и место отправления результатов. Т. е. данная нотация сфокусирована не столько на процессе, сколько на движении информационного потока.

Как и любая другая нотация, DFD имеет достоинства и недостатки.

Главным преимуществом нотации является отсутствие требований к соблюдению правил и синтаксиса. Нотация не является исполнимой, она используется для формирования внутреннего понимания, структурирования и последующей работы с данными.

Кроме того, важными положительными характеристиками являются:

1. Способность к точному определению внешних сущностей при использовании методов анализа потоков информации в системе и за ее пределами.
2. Возможность вертикального проектирования.
3. Обеспечение возможности описания процессов нижних уровней, что позволяет достичь логической завершенности модели.

Основным недостатком называют отсутствие возможности анализа временных параметров в модели, необходимость использования управляющих процессов.

Символы и синтаксис диаграмм

Исторически были разработаны два варианта синтаксиса нотации: Йордана и Гейна-Сарсона. В этих вариантах имеются различия в отображении процесса (в первом варианте процесс отображается кругом, а во втором – разделенным горизонтальной чертой прямоугольником), а также хранилища данных (в варианте Йордана хранилище данных представлено простым прямоугольником, а в варианте Гейна – прямоугольник имеет разделяющую вертикальную черту).

Нотация включает в себя 4 основных элемента:

1. Процесс, который может содержать в себе функцию или последовательность действий, необходимых для обработки информации. В названии процесса преимущественно используют глаголы.
2. Внешние сущности, представленные любыми объектами, не включенными в систему, но поставляющие или получающие информацию из системы.
3. Хранилище данных является внутренним объектом. Это может быть базой данных, таблицей или любым другом вариантом, в котором организовано хранение данных.
4. Поток данных отображается стрелкой, которая показывает входящую и исходящую из каждого блока информацию.

Уровни и слои

Уровни и слои в DFD дают возможность наполнить ее значительным количеством подробностей. Для этого внимание фокусируется на конкретном участке. Уровень диаграммы обозначается цифрой от нуля.

Диаграмма нулевого уровня – это контекстная схема. На ней отображаются анализируемые или моделируемые системы. На схеме отображена общая картина, в которой система представлена единым процессом, связанным с внешними сущностями. Схема нулевого уровня понятна любому неподготовленному читателю.

На схеме первого уровня уже более детально представлены элементы, отображенные на контекстной схеме. Здесь общий процесс разбит на подпроцессы и выделены основные функции.

Схема второго уровня еще более глубоко погружает аналитика в сущность системы. Однако для более полного и подробного описания схема должна содержать большее количество текстовой информации.

Дальнейшая детализация также возможно, но далее третьего уровня составители редко заходят, так как это усложняет понимание модели, поэтому снижается уровень эффективности в решении задач, связанных со сравнением, моделированием и передачей информации.

Использование слоев дает возможность сбора выпадающих уровней непосредственно в схеме. Это позволяет достичь симбиоза глубокого анализа с ясностью изложения.

Высокий уровень детализации дает возможность составления псевдокода, в котором сочетаются программный и естественный язык. С помощью псевдокода упрощается работа по написанию полноценного кода.

Логические и физические диаграммы

Диаграммы рассматриваемой нотации могут быть представлены в двух категориях – логические и физические. В логических диаграммах наглядно показано то, как перемещаются данные, имеющие жизненную важность для работы предприятия. Центр такой диаграммы – непосредственно бизнес, а не особенности работы системы.

Физические диаграммы дают иллюстрацию того, как система устроена на данный момент или какого состояния должна достигнуть в будущем. Так, логическая диаграмма в качестве процессов будет рассматривать различные виды работ организации, а физическая – программы и ручные процедуры.

Примеры применения схем потоков данных

Основной сферой применения DFD-диаграммы является разработка программного обеспечения. В этой сфере под хранилищем данным понимают различные электронные таблицы и базы данных; внешними сущностями будут являться клиенты или другие базы данных; под процессами понимают модули и функции, которые выполняются в системе.

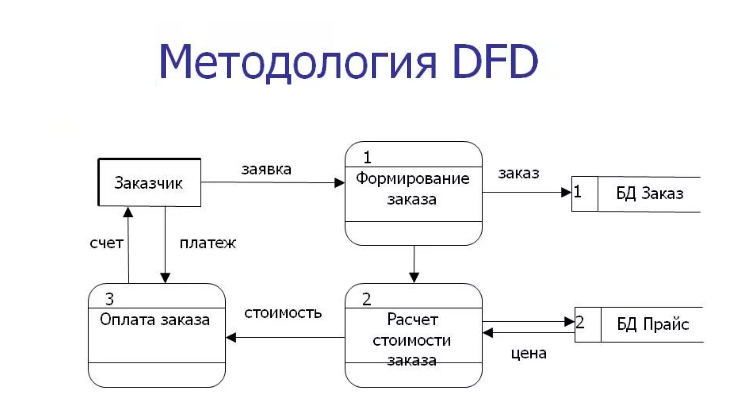


Рисунок 4. Методология DFD

Также удобно использование данных диаграмм при проведении анализа организации документооборота. Использование DFD в данном случае позволяет получить понимание мест хранения данных, способов обмена документацией, определить точки, в которых допущены ошибки и неправильно организованы процессы.

Но в этом случае нотацию нужно использовать с некоторой осторожностью, т.к. непосредственно бизнес-процесс в ней не описывается. Тем не менее, есть положительные результаты использования нотации в качестве вспомогательного варианта и демонстрации того, какие проблемы выявлены и каким методом возможно их устранение.

**1.6 Методика моделирования потоков данных**

Целью методики является построение модели рассматриваемой системы в виде диаграммы потоков данных (Data Flow Diagram – DFD), обеспе- Вводное занятие 19 чивающей правильное описание выходов (отклика системы в виде данных) при заданном воздействии на вход системы (подаче сигналов через внешние интерфейсы). Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

При создании диаграммы потоков данных используются четыре основных понятия: потоки данных, процессы (работы) преобразования входных потоков данных в выходные, внешние сущности, накопители данных (хранилища). Потоки данных являются абстракциями, использующимися для моделирования передачи информации (или физических компонент) из одной части системы в другую.

Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации. Назначение процесса (работы) состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, «получить документы по отгрузке продукции»).

Каждый процесс имеет уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели. Хранилище (накопитель) данных позволяет на указанных участках определять данные, которые будут сохраняться в памяти между процессами.

Фактически хранилище представляет «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее получения, при этом данные могут выбираться в любом порядке. Имя хранилища должно определять его содержимое и быть существительным. Внешняя сущность представляет собой материальный объект вне контекста системы, являющейся источником или приемником системных данных. Ее имя должно содержать существительное, например, «склад товаров». Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке.

Кроме основных элементов, в состав DFD входят словари данных и миниспецификации. Словари данных являются каталогами всех элементов данных, присутствующих в DFD, включая групповые и индивидуальные потоки данных, хранилища и процессы, а также все их атрибуты.

Миниспецификации обработки – описывают DFD-процессы нижнего уровня. Фактически миниспецификации представляют собой алгоритмы описания задач, выполняемых процессами: множество всех миниспецификаций является полной спецификацией системы. Процесс построения DFD начинается с создания так называемой основной диаграммы типа «звезда», на которой представлен моделируемый

Вводное занятие 20 процесс и все внешние сущности, с которыми он взаимодействует. В случае сложного основного процесса он сразу представляется в виде декомпозиции на ряд взаимодействующих процессов. Критериями сложности в данном случае являются: наличие большого числа внешних сущностей, многофункциональность системы, ее распределенный характер. Внешние сущности выделяются по отношению к основному процессу.

Для их определения необходимо выделить поставщиков и потребителей основного процесса, т.е. все объекты, которые взаимодействуют с основным процессом. На этом этапе описание взаимодействия заключается в выборе глагола, дающего представление о том, как внешняя сущность использует или используется основным процессом. Например, основной процесс – «учет обращений граждан», внешняя сущность – «граждане», описание взаимодействия – «подает заявления и получает ответы».

Этот этап является принципиально важным, поскольку именно он определяет границы моделируемой системы. Для всех внешних сущностей строится таблица событий, описывающая их взаимодействие с основным потоком. Таблица событий включает в себя наименование внешней сущности, событие, его тип (типичный для системы или исключительный, реализующийся при определенных условиях) и реакцию системы. На следующем шаге происходит декомпозиция основного процесса на набор взаимосвязанных процессов, обменивающихся потоками данных. Сами потоки не конкретизируются, определяется лишь характер взаимодействия. Декомпозиция завершается, когда процесс становится простым, т.е.

1. процесс имеет два-три входных и выходных потока
2. процесс может быть описан в виде преобразования входных данных в выходные
3. процесс может быть описан в виде последовательного алгоритма

Для простых процессов строится миниспецификация – формальное описание алгоритма преобразования входных данных в выходные.

Миниспецификация удовлетворяет следующим требованиям: для каждого процесса строится одна спецификация; спецификация однозначно определяет входные и выходные потоки для данного процесса; спецификация не определяет способ преобразования входных потоков в выходные; спецификация ссылается на имеющиеся элементы, не вводя новые; спецификация по возможности использует стандартные подходы и операции.

После декомпозиции основного процесса для каждого подпроцесса строится аналогичная таблица внутренних событий. Следующим шагом после определения полной таблицы событий выделяются потоки данных, которыми обмениваются процессы и внешние сущности. Простейший способ их выделения заключается в анализе таблиц событий.

События преобразуются в потоки данных от инициатора события к запрашиваемому процессу, а реакции – в обратный поток событий. После Вводное занятие 21 построения входных и выходных потоков аналогичным образом строятся внутренние потоки. Для их выделения для каждого из внутренних процессов выделяются поставщики и потребители информации. Если поставщик или потребитель информации представляет процесс сохранения или запроса информации, то вводится хранилище данных, для которого данный процесс является интерфейсом.

После построения потоков данных диаграмма должна быть проверена на полноту и непротиворечивость. Полнота диаграммы обеспечивается, если в системе нет «повисших» процессов, не используемых в процессе преобразования входных потоков в выходные. Непротиворечивость системы обеспечивается выполнением наборов формальных правил о возможных типах процессов: на диаграмме не может быть потока, связывающего две внешние сущности – это взаимодействие удаляется из рассмотрения; ни одна сущность не может непосредственно получать или отдавать информацию в хранилище данных – хранилище данных является пассивным элементом, управляемым с помощью интерфейсного процесса; два хранилища данных не могут непосредственно обмениваться информацией – эти хранилища должны быть объединены.

К преимуществам методики DFD относятся: возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы; возможность проектирования сверху вниз, что облегчает построение модели «как должно быть»; наличие спецификаций процессов нижнего уровня, что позволяет преодолеть логическую незавершенность функциональной модели и построить полную функциональную спецификацию разрабатываемой системы. К недостаткам модели отнесем: необходимость искусственного ввода управляющих процессов, поскольку управляющие воздействия (потоки) и управляющие процессы с точки зрения

DFD ничем не отличаются от обычных; отсутствие понятия времени, т.е. отсутствие анализа временных промежутков при преобразовании данных (все ограничения по времени должны быть введены в спецификациях процессов). Модель DFD строится как самостоятельная модель параллельно с моделью, выполняемой в нотации IDEF0.

**Глава 2. Процесс создания блок-схем в Ramus**

**2.1 Методология IDEF0. создание контекстной диаграммы**

Перед началом работы необходимо ознакомится с материалами лекции «Информационное моделирование экономических процессов при создании ИС», а также с руководящим документом (стандартом) Методология функционального проектирования IDEF0

Сведения о проектируемой системе

В лабораторной работе речь пойдет о построении диаграмм при проектировании системы «Служба занятости в рамках ВУЗа».

Поскольку процесс проектирования системы начинается с изучения предметной области, опишем данную предметную область здесь.

Разрабатываемая система предназначена для того, чтобы помочь студенту устроиться на работу уже в процессе его обучения в ВУЗе. Подав заявление в систему студент становится её клиентом и начинает обслуживаться на протяжении всего обучения в Вузе. Система предлагает профессиональные (основанные на изучаемых предметах) и психологические тестирования. Тестирования проводятся регулярно – раз в семестр.

Особое внимание уделяется обучению студента. Информация об успеваемости заносится в систему. По итогам успеваемости и результатам тестирования составляются экспертные оценки. Кроме того, система позволяет студенту формировать и хранить резюме.

**2.1 Методология IDEF0. создание контекстной диаграммы**

При запуске системы Ramus Educational появляется окно вида:

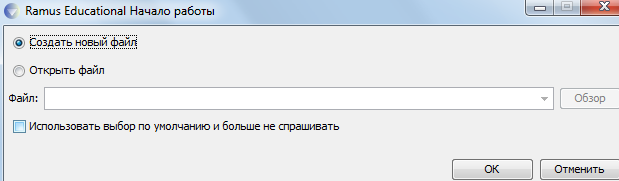


Рисунок 5. Пример создания функциональной модели DFD

1. Выберем опцию Создать новый файл и нажмем кнопку ОК
2. В появившемся Мастере свойств проекта внесите:
3. Автор – свое имя
4. Название проекта – Служба занятости в рамках ВУЗа
5. Название модели – не пишем
6. Выберите нотацию IDEF0
7. После заполнения необходимых сведений нажмите кнопку и перейдите к следующему шагу.
8. Укажите, что модель используется в Управление кадров. Студенческий отдел

На следующем шаге в описании проекта укажите: Учебная модель функционирования системы Служба занятости ВУЗа Цель моделирования – подготовить описание функционирования системы службы занятости, которое было бы понятно её пользователю, не вдаваясь в подробности, связанные с реализацией.

Раздел "классификаторы" оставьте незаполненным и нажмите

В следующем диалоговом окне нажмите и перейдите к рабочему интерфейсу программы.

В дальнейшем через меню Диаграмма -> Свойства модели можно отредактировать мета-данные модели, а именно: название модели, описание, место ее использования.

Активируйте окно модели, кликнув на область моделирования.

Начнем с построения контекстную диаграмму, нажав на кнопку http:контекстной IDEF0-диаграммы.

Согласно описанию системы основной её функцией является обслуживание клиентов посредством обработки запросов, от клиентов поступающих. определим единственную работу контекстной диаграммы, как «Обслужить клиента системы».

1. Перейдите в режим редактирования контекстной диаграммы, нажав правой кнопкой мыши на объекте и выбрав опцию "Редактировать активный элемент". Или щелкнув двойным щелчком левой кнопки мыши по объекту.
2. В появившемся окне Свойства функционального блока в закладке "Название" введите "Обслуживание клиента системы".

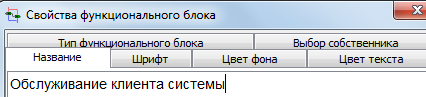


Рисунок 6. Пример создания функциональной модели DFD

При необходимости растяните функциональный блок контекстной диаграммы до устраивающих вас размеров.

Далее определим входные и выходные данные, а также механизмы и управление. Для того, чтобы обслужить клинта необходимо зарегистрировать его в системе, открыть доступ к БД и обработать его запрос. , в качестве входных данных будут использоваться Имя клиента, Пароль клиента, Исходная БД, Запрос клиента.

Для создания стрелок необходимо перейти в режим построения стрелок с помощью кнопки http: jpg, навести курсор на исходную точку стрелки (левая, верхняя и нижняя граница области построения модели или правая граница контекстной диаграммы), после того, как область будет подсвечена черным цветом, кликнуть один раз и аналогичным образом обозначить конец стрелки (правая, верхняя и нижняя граница контекстной диаграммы или правая граница области построения модели).

Для того, чтобы дать стрелке имя, также как и в случае с функциональным блоком необходимо щелкнуть двойным щелчком левой кнопки мыши по стрелке, или же вызвав щелчком по правой кнопке мыши при указании на стрелку контекстное меню выбрать в нем пункт Редактировать активный элемент. Перемещать стрелки и их названия можно по принципам стандартного механизма drag&drop.

Выполнение запроса ведет либо к получению информации из системы, либо к изменению содержимого БД (например, при составлении экспертных оценок), поэтому выходными данными будут являться Отчеты и Измененная БД.

Процесс обработки запросов будет выполнять Пользователь системы под контролем Администратор системы

Управляющей стрелкой будут являться Уровни доступа.

Результат построения контекстной диаграммы будет иметь вид:

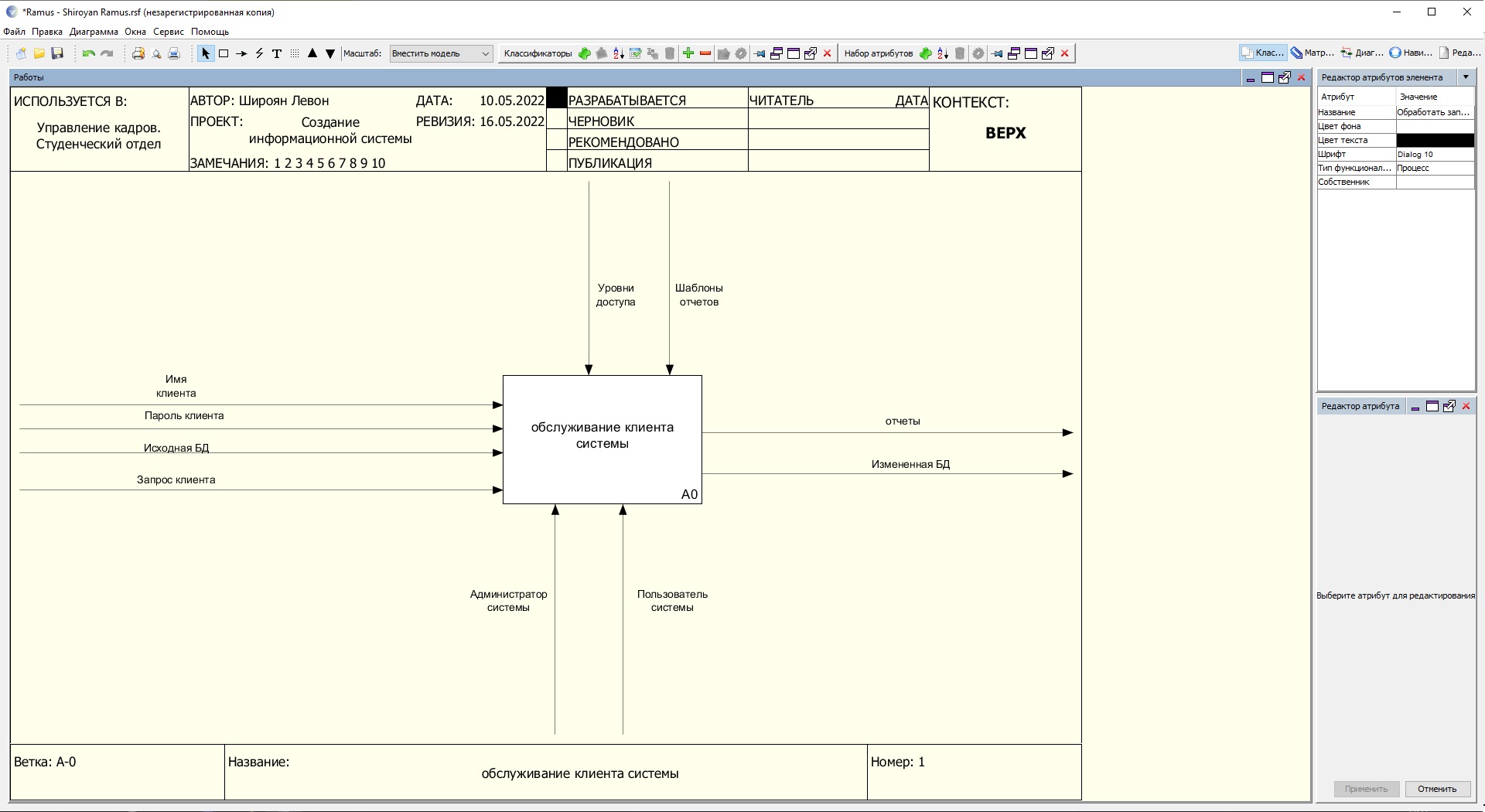


Рисунок 7. Пример создания функциональной модели DFD

**2.2 Методология IDEF0. создание декомпозиции контекстной диаграммы**

Проведем декомпозицию контекстной диаграммы описав последовательность обслуживания клиентов:

Выберите кнопку перехода на уровень ниже http: jpgв панели инструментов.

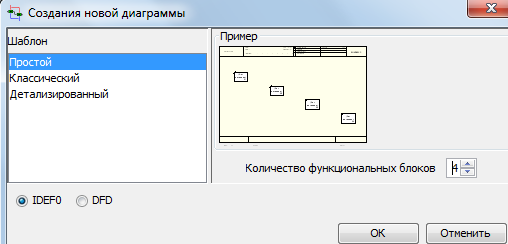


Рисунок 8. Пример создания функциональной модели DFD

В диалоговом окне укажите число работ на диаграмме нижнего уровня - "4", а нотацию декомпозиции – IDEF0, затем нажмите "ОК". Автоматически будет создана диаграмма декомпозиции.

Правой кнопкой мыши щелкните по 1-ой работе, выберите "Редактировать активный элемент" и на вкладке "Название" укажите имя работы. Повторите операцию для всех четырех работ, используя следующую последовательность обслуживания клиента:

1. Определение уровня доступа в систему
2. Обращение к подсистеме
3. Изменение БД (при необходимости)
4. Обработка запроса клиента

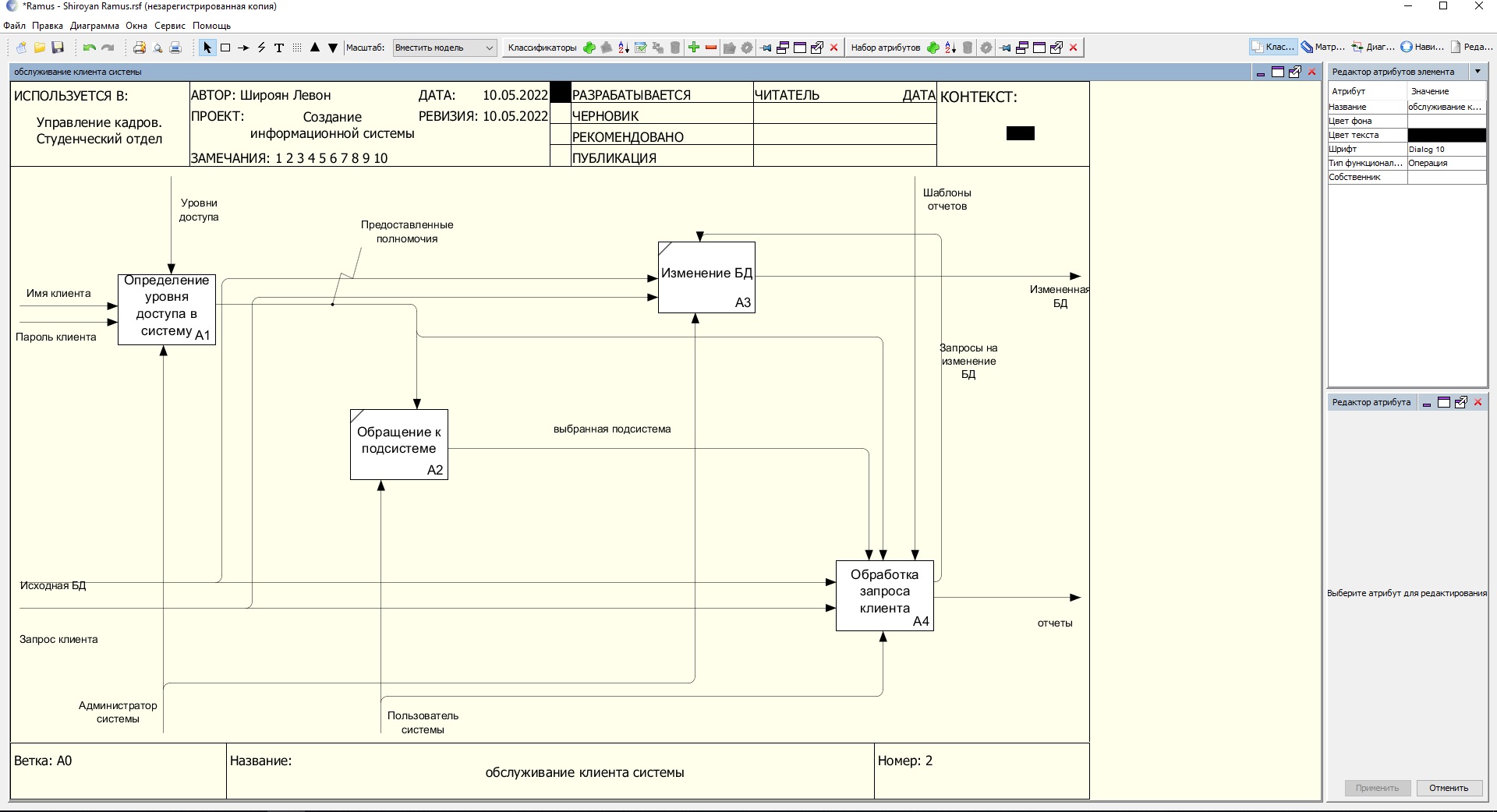


Рисунок 9. Пример создания функциональной модели DFD

При необходимости для улучшения визуального восприятия диаграммы растяните функциональные блоки. Обратите внимание на стрелки, перешедшие с родительской диаграммы, на дочернюю.

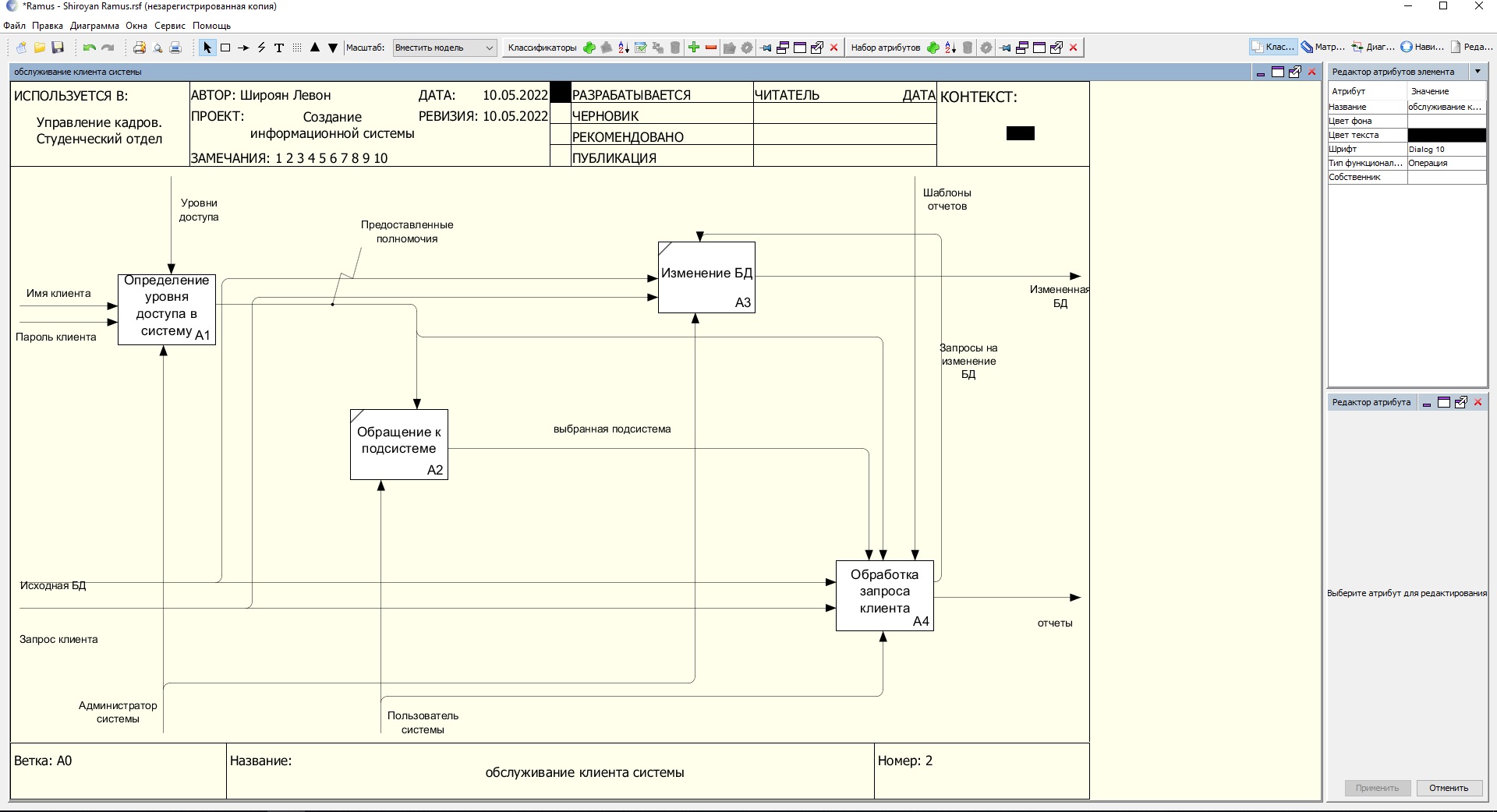


Рисунок 10. Пример создания функциональной модели DFD

Перейдите в режим рисования стрелок. Произведите связывание граничных стрелок с функциональными объектами, как показано на нижеприведенном рисунке. Для связывания граничных стрелок наводите курсор на сами стрелки, а не на границы области построения моделей. Не забудьте создать новые внутренние стрелки.

Обязательно создайте стрелку обратной связи (по управлению) Запросы на изменение БД, идущую от работы Обработка запроса клиента к Изменение БД. Изменить стиль стрелки – толщину (правая кнопка мыши – Редактировать активный элемент – вкладка Линия)

Методом drag&drop возможно переносить стрелки и их названия. При необходимости возможно установить "тильду" (опция контекстного меню при нажатии на стрелке правой кнопкой мыши) для явной связи стрелки и подписи к ней.

В результате получится представление вида:

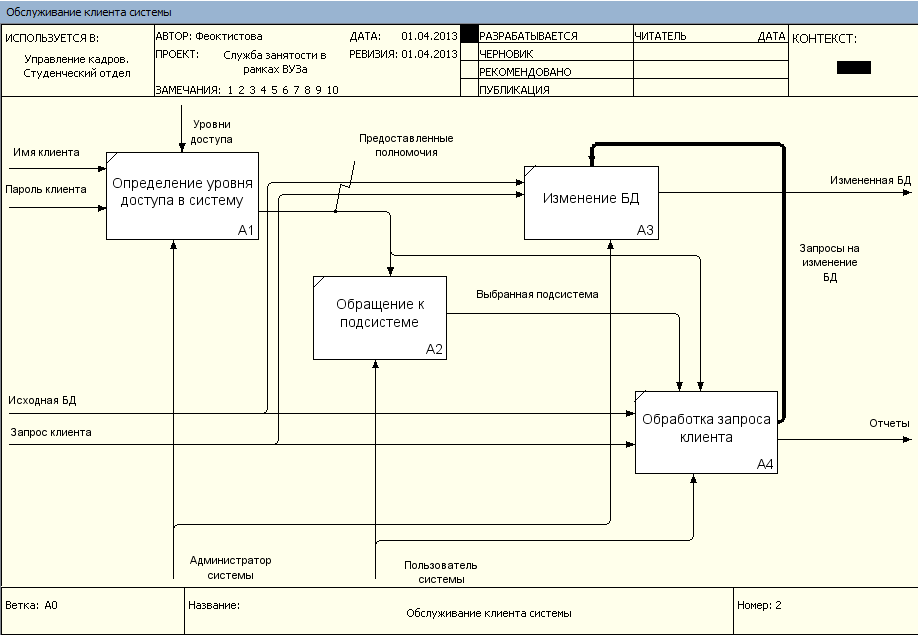


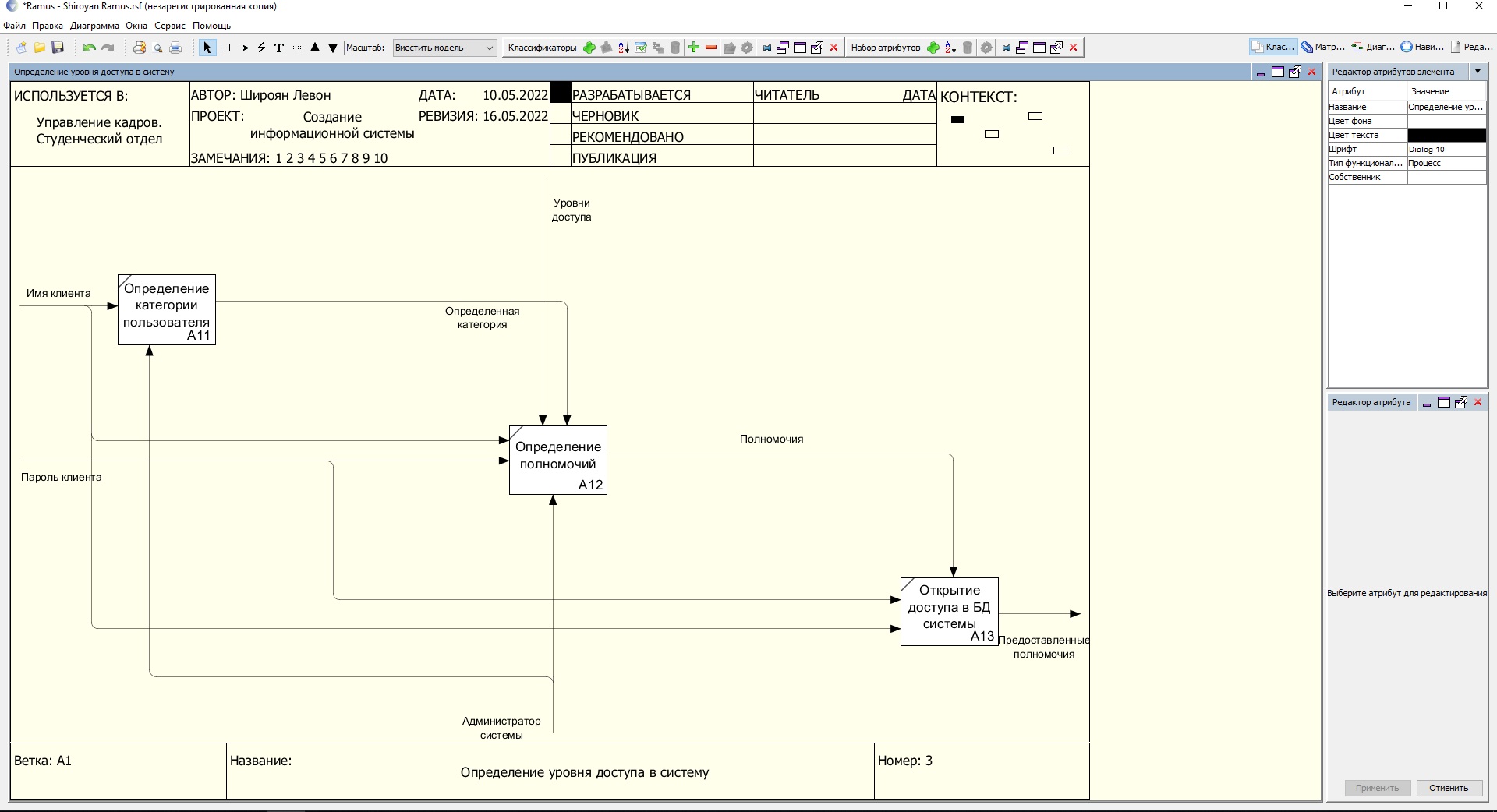
Рисунок 11. Пример создания функциональной модели DFD

**2.3 Методология IDEF0. создание дальнейших диаграмм декомпозиций**

Закончив декомпозицию контекстной диаграммы, переходят к декомпозиции диаграммы следующего уровня. Обычно, при рассмотрении третьего и более нижних уровней модели возвращаются к родительским диаграммам и корректируют их.

Декомпозируем последовательно все блоки полученной диаграммы.

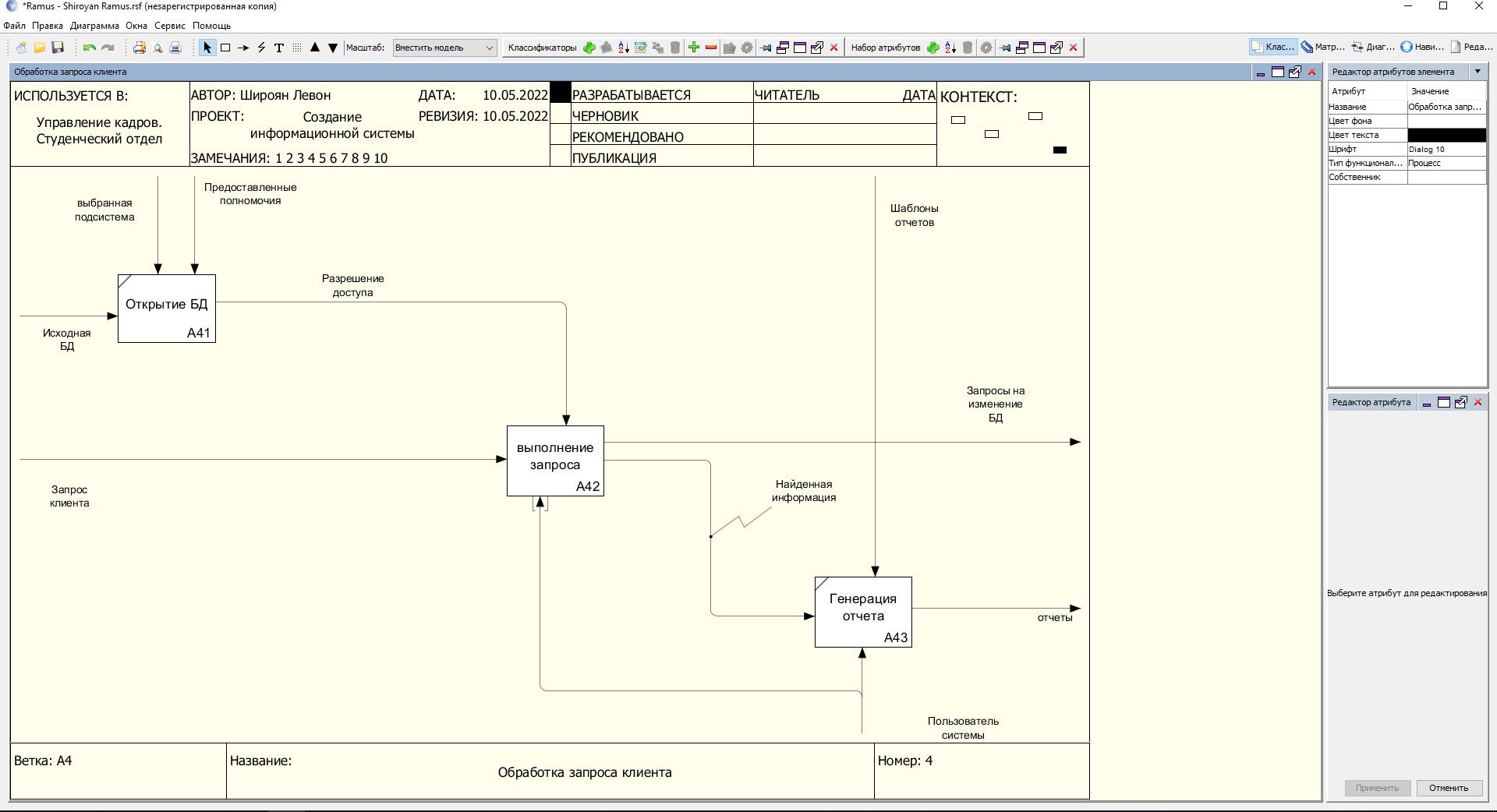
Рисунок 12. Пример создания функциональной модели DFD



Начнем с блока Определение уровня доступа в систему. Первым этапом при определении уровня доступа в систему является определение категории пользователя. По имени клиента осуществляется поиск в базе пользователей (студент, преподаватель, фирма, деканат, сотрудник отдела кадров) и определяется его категория. Согласно категории выясняются полномочия, предоставляемые пользователю системы. Далее проводиться процедура доступа в систему (проверка имени и пароля доступа). Объединяя информацию о полномочиях и уровне доступа в систему для пользователя формируется набор разрешенных действий.

Начиная декомпозицию блока: Определение уровня доступа в систему выделим этот функциональный блок на декомпозиции первого уровня и нажмем кнопку определение уровня доступа в систему будет выглядеть следующим образом:

Рисунок 13. Пример создания функциональной модели DFD



Декомпозиция работы Обращение к подсистеме не отвечает цели и точки зрения модели. Пользователя системы не интересуют внутренние алгоритмы её работы. Поэтому декомпозиция данного блока не проводится. Аналогично обстоит дело с работой Изменение БД.

В дальнейшей декомпозиции нуждается блок Обработка запроса клиента. Декомпозируя работу будем учитывать, что перед осуществлением поиска ответа на запрос клиента необходимо сообщить системе об установлении соединения с БД, после чего выполнить запрос и сгенерировать отчеты для пользователя. Поэтому последовательность работ у нас будет следующей:

Необходимо отметить, что в блок Выполнение запроса включается работа различных подсистем. Например, если запрос включает в себя тестирование, то его будет исполнять подсистема профессиональных и психологических тестов. При анализе полученной диаграммы возникает вопрос по каким правилам проводится генерация отчетов? Необходимо наличие заранее сформированных шаблонов, по которым будет производиться выборка из БД. Причем эти шаблоны должны соответствовать запросам и должны быть заранее определены. Клиенту же предоставляется возможность выбора формы отчета.

Скорректируем диаграмму, добавив в неё стрелку Шаблоны отчетов.

Эта стрелка автоматически не попадает в диаграмму верхнего уровня и имеет квадратные скобки у окончания (тунелирование), поэтому необходимо щелкнуть правой кнопкой мыши по квадратным скобкам и выбрать в контекстном меню пункт Тунель. Система предложит остановиться на одной из двух опций:

Тунелирование применяется для того, чтобы не выносить на верхнеуровневые диаграммы малозначимые взаимодействия, которыми можно пренебречь.

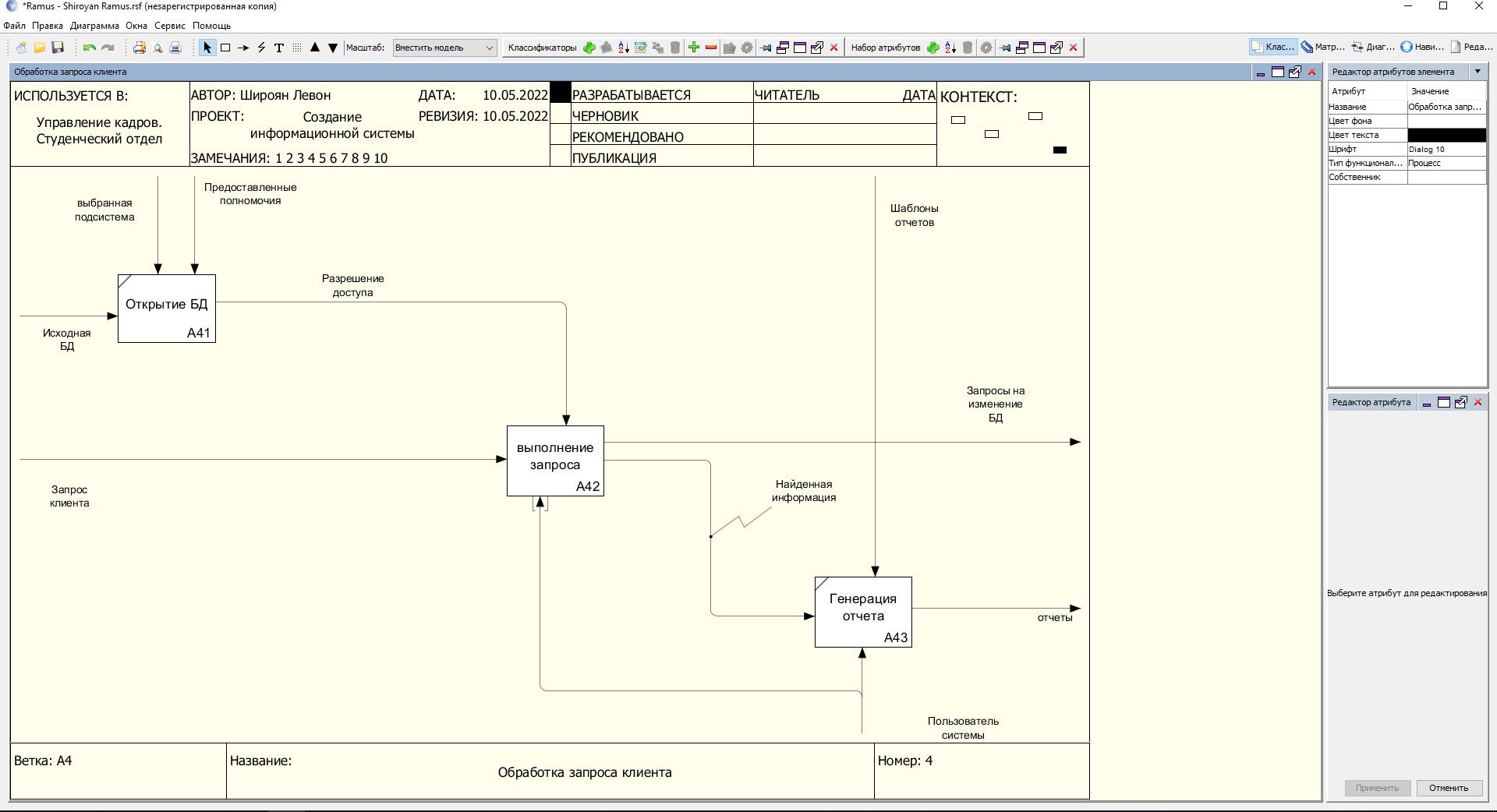


Рисунок 14. Пример создания функциональной модели DFD

В нашем случае выберем первый вариант. изменение диаграммы повлечет за собой корректировку всех родительских диаграмм.

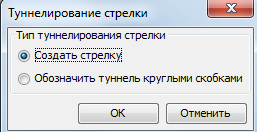


Рисунок 15. Пример создания функциональной модели DFD

Декомпозиция процесса Обработка запроса клиента после корректировки примет вид:

Вернемся на диаграммы верхних уровней и сделаем необходимые изменения:

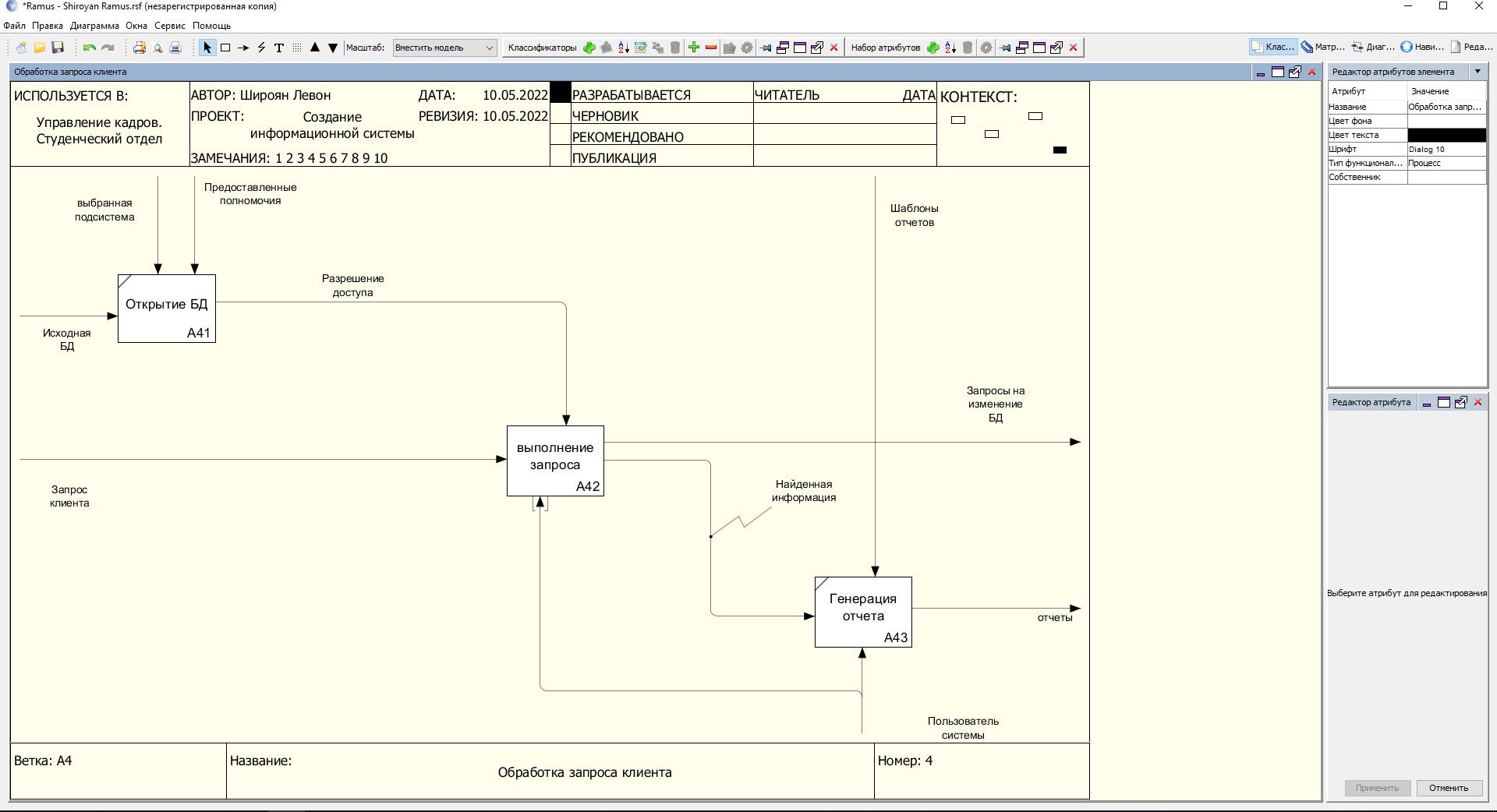


Рисунок 16. Пример создания функциональной модели DFD

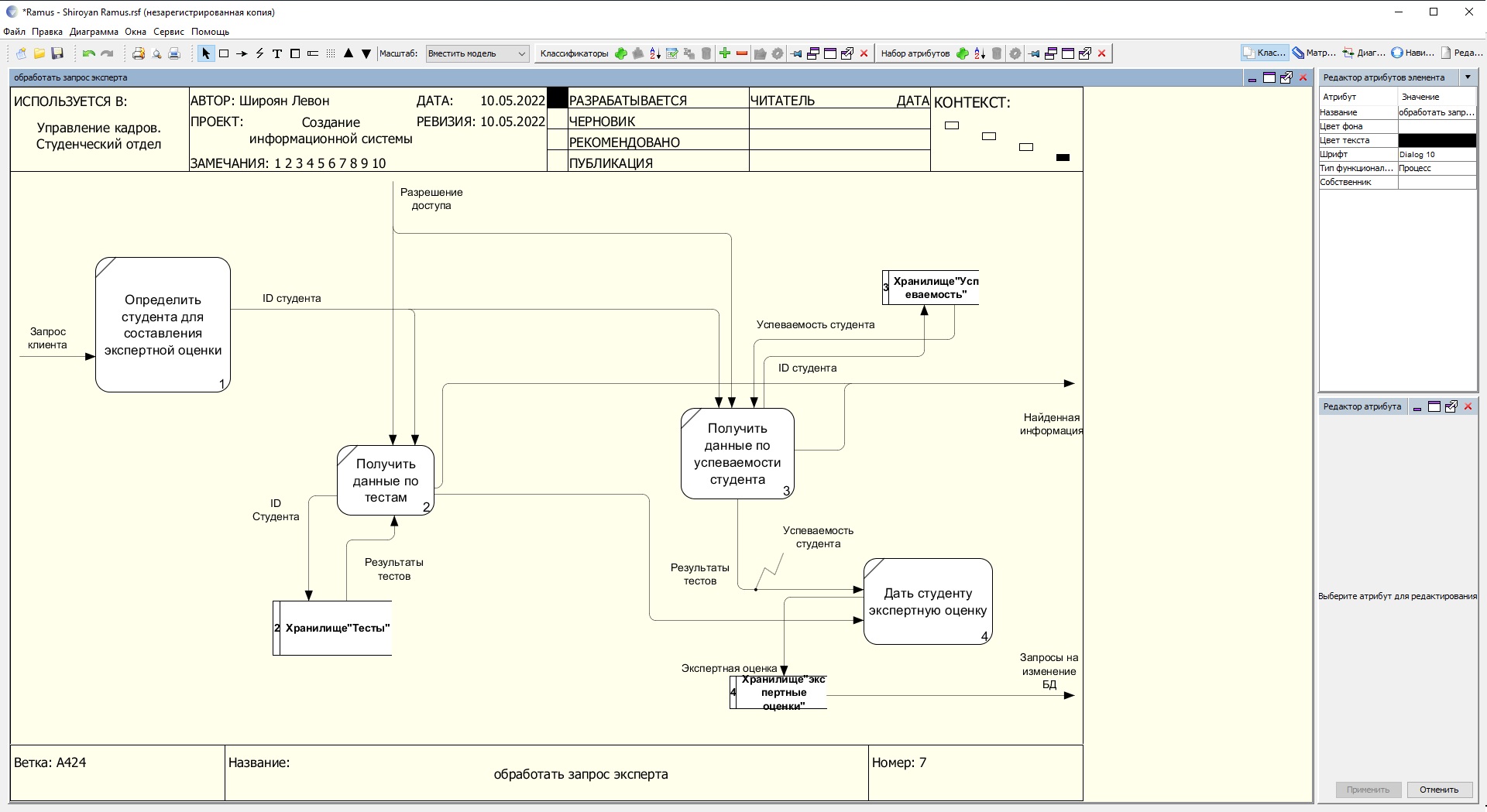


Рисунок 17. Пример создания функциональной модели DFD

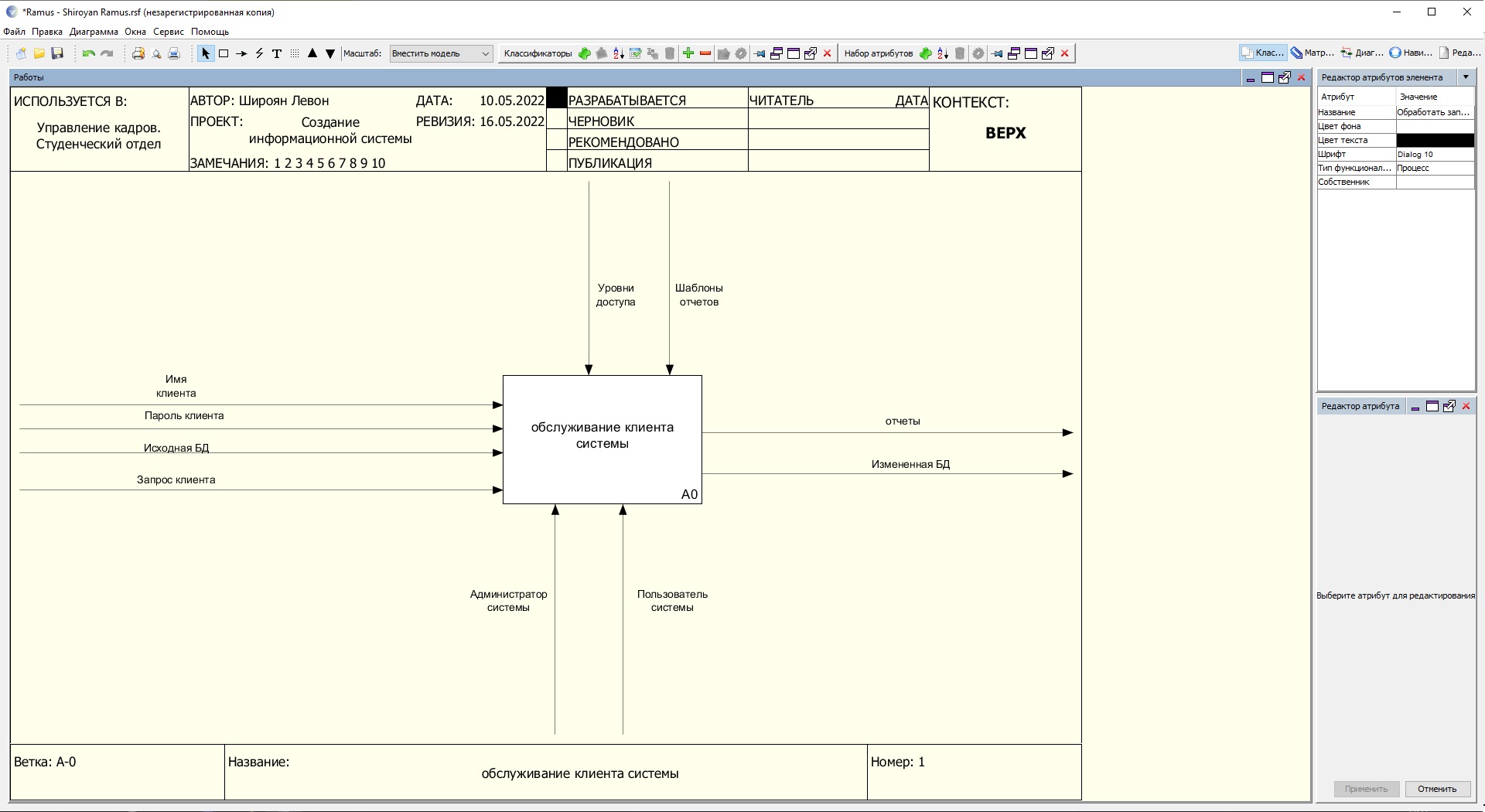


Рисунок 18. Пример создания функциональной модели DFD

**2.4 Методология DFD. понятийный аппарат**

Диаграммы потоков данных (DFD, Data Flow Diagram) – представляют собой сеть связанных между собой работ. Их удобно использовать для описания документооборота и обработки информации.

DFD описывает:

1. Функции обработки информации – работы
2. Документы, объекты, сотрудников или отделы, участвующие в процессе обработки информации
3. Внешние ссылки, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы
4. Таблицы для хранения документов (хранилища данных)

Для построения диаграмм DFD в Ramus Educational используются следующие обозначения

1. Поток данных
2. Внешняя сущность

Потоки данных являются механизмами, использующимися для моделирования передачи информации (или физических компонентов) из одной части системы в другую. Потоки изображаются на схеме именованными стрелками, ориентация которых указывает направление движения информации. Стрелки могут подходить к любой грани работы и могут быть двунаправленными для описания взаимодействия типа команда-ответ.

Назначение процесса состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы.

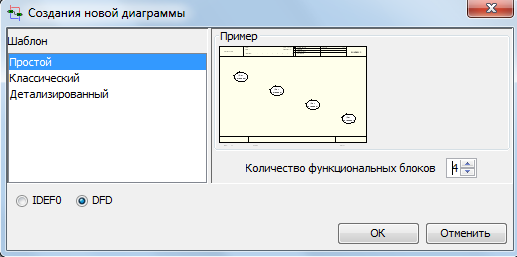


Рисунок 19. Создание новой диаграммы

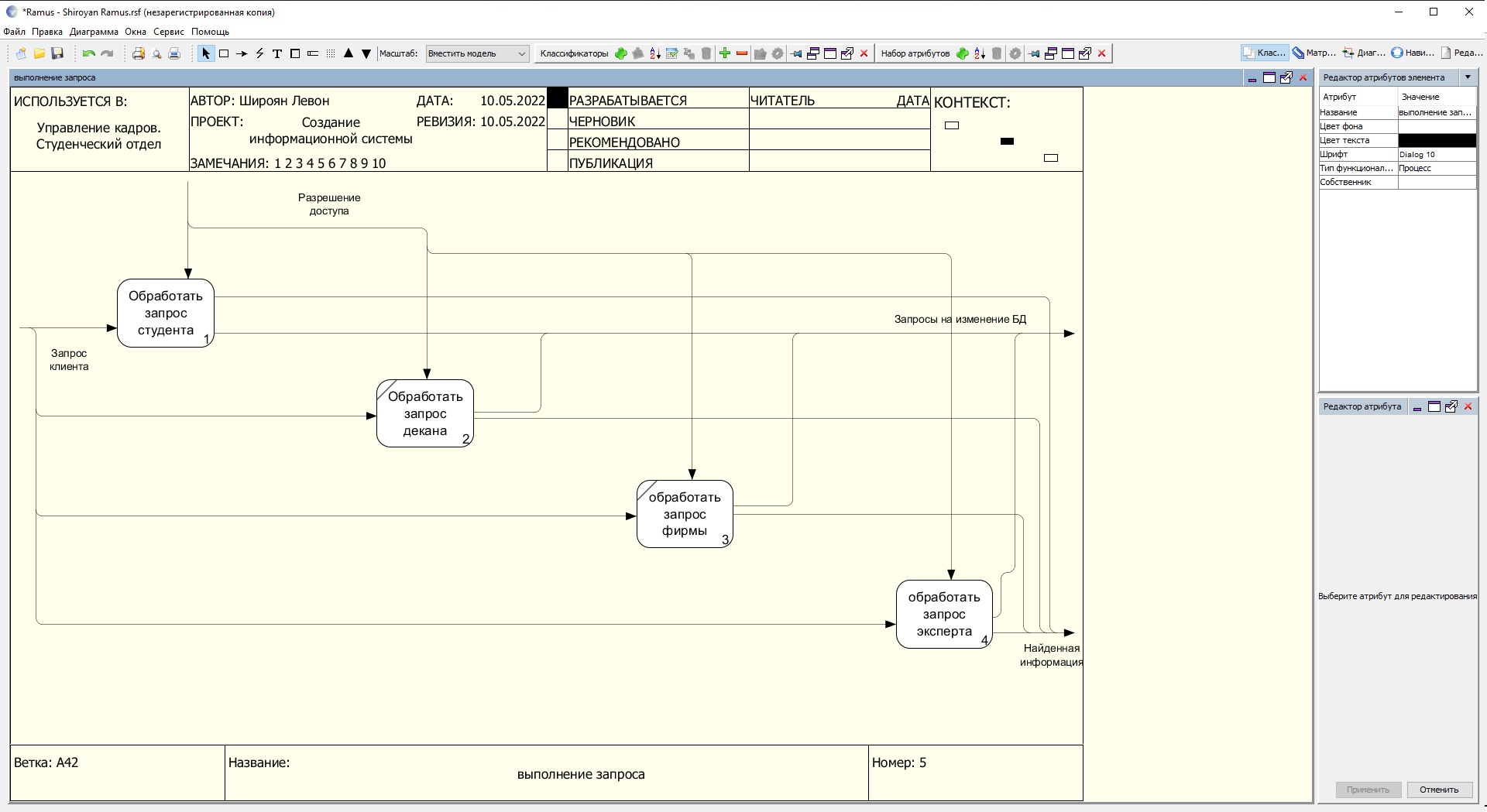


Рисунок 20. Пример создания функциональной модели DFD

Хранилище данных позволянет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилища – это «срезы» потоков данных во времени. Информация, содержащаяся в хранилище может использоваться в любое время после её определения при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое.

Внешняя сущность представляет собой сущность вне котекста системы, являющуюся источником или приемником даны системы. Предполагается, что объекты представленные внешними сущностями не должны участвовать ни в какой обработке. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах.

**2.5 Дополнение моделей процессов диаграммой DFD**

Декомпозицию работы Выполнение запроса целесообразно провести при помощи диаграммы DFD. методология IDEF0 рассматривает систему, как совокупность взаимосвязанных работ, что плохо отражает процессы обработки информации.

Выбрав работу Выполнение запроса с помощью кнопки начнем построение дочерней диаграммы в нотации DFD.

Все процессы обработки запросов контролируются и выполняются внутренними механизмами системы, при участии пользователя, поэтому стрелка-механизм Пользователь системы будет повторяться на декомпозированных диаграммах. Точка зрения модели, определенная ранее, не требует рассмотрения внутренних особенностей функционирования системы, поэтому затунелируем стрелку Пользователь системы с тем, чтобы не переносить её на диаграммы нижних уровней.

Тунелирование производится удалением стрелки Пользователь системы с текущей декомпозиции.

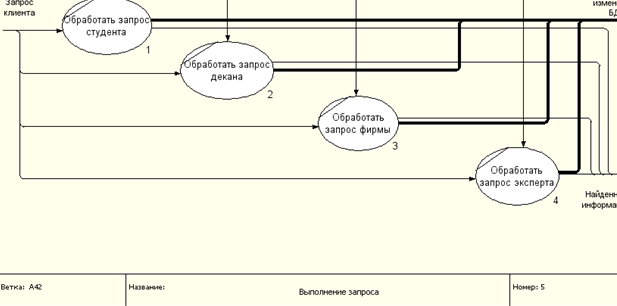


Рисунок 21. Пример создания функциональной модели DFD

Родительская диаграмма примет вид:

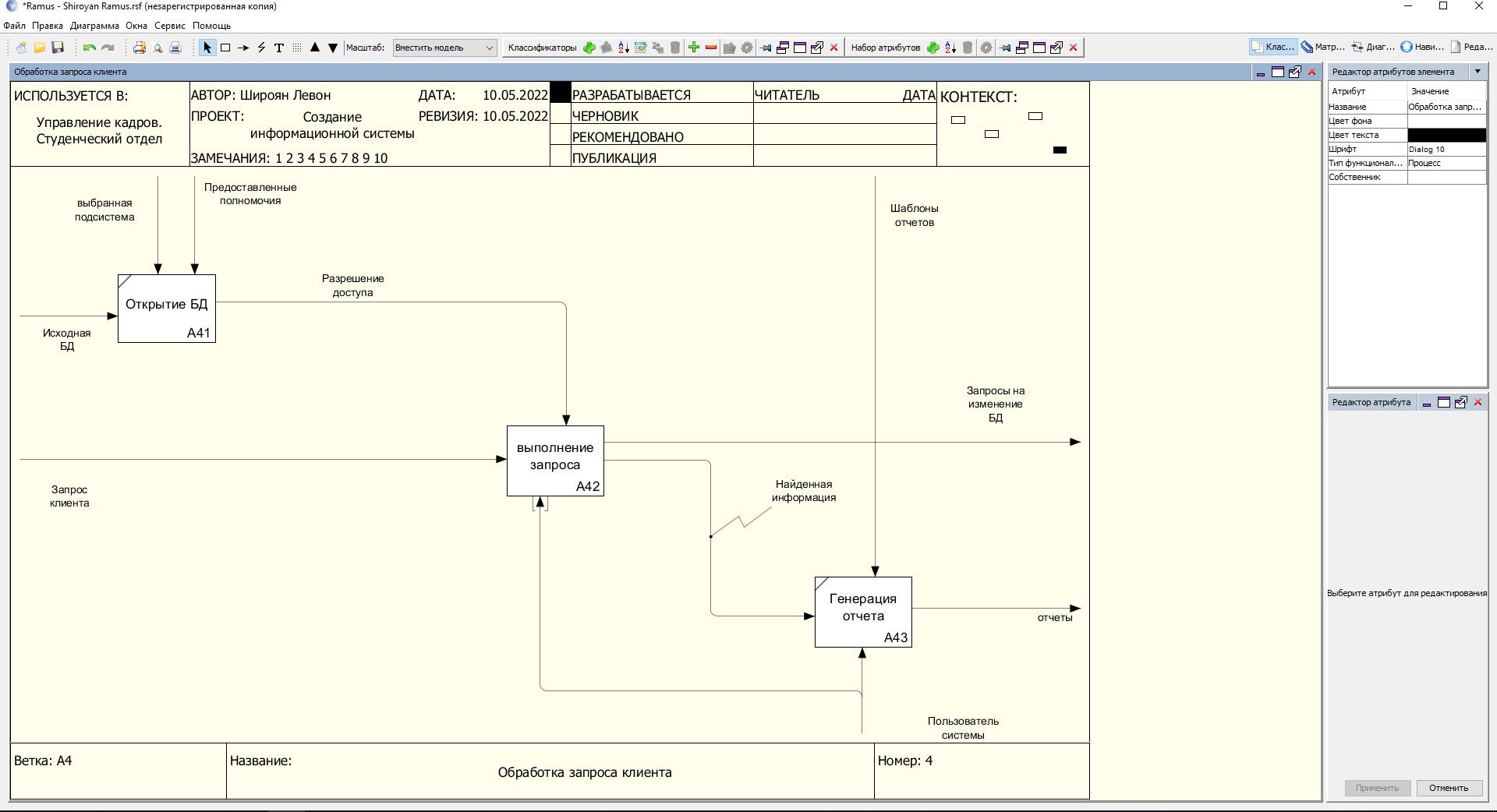


Рисунок 22. Пример создания функциональной модели DFD

Декомпозируем работу Обработать запрос студента в нотации DFD.

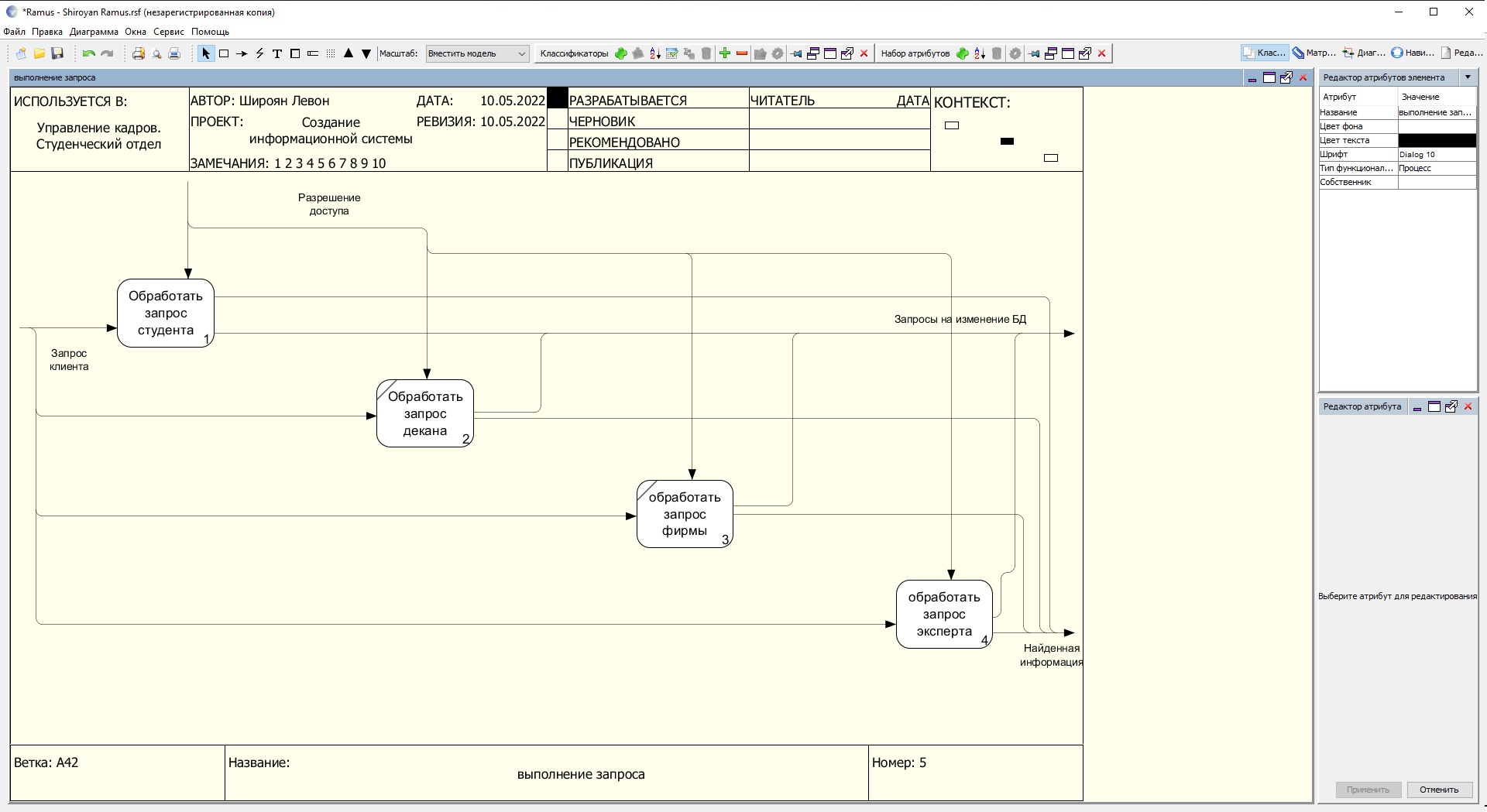


Рисунок 23. Пример создания функциональной модели DFD

При формировании декомпозиции внесите в диаграмму имена работ:

1. Составить резюме
2. Найти данные по успеваемости
3. Пройти тестирование

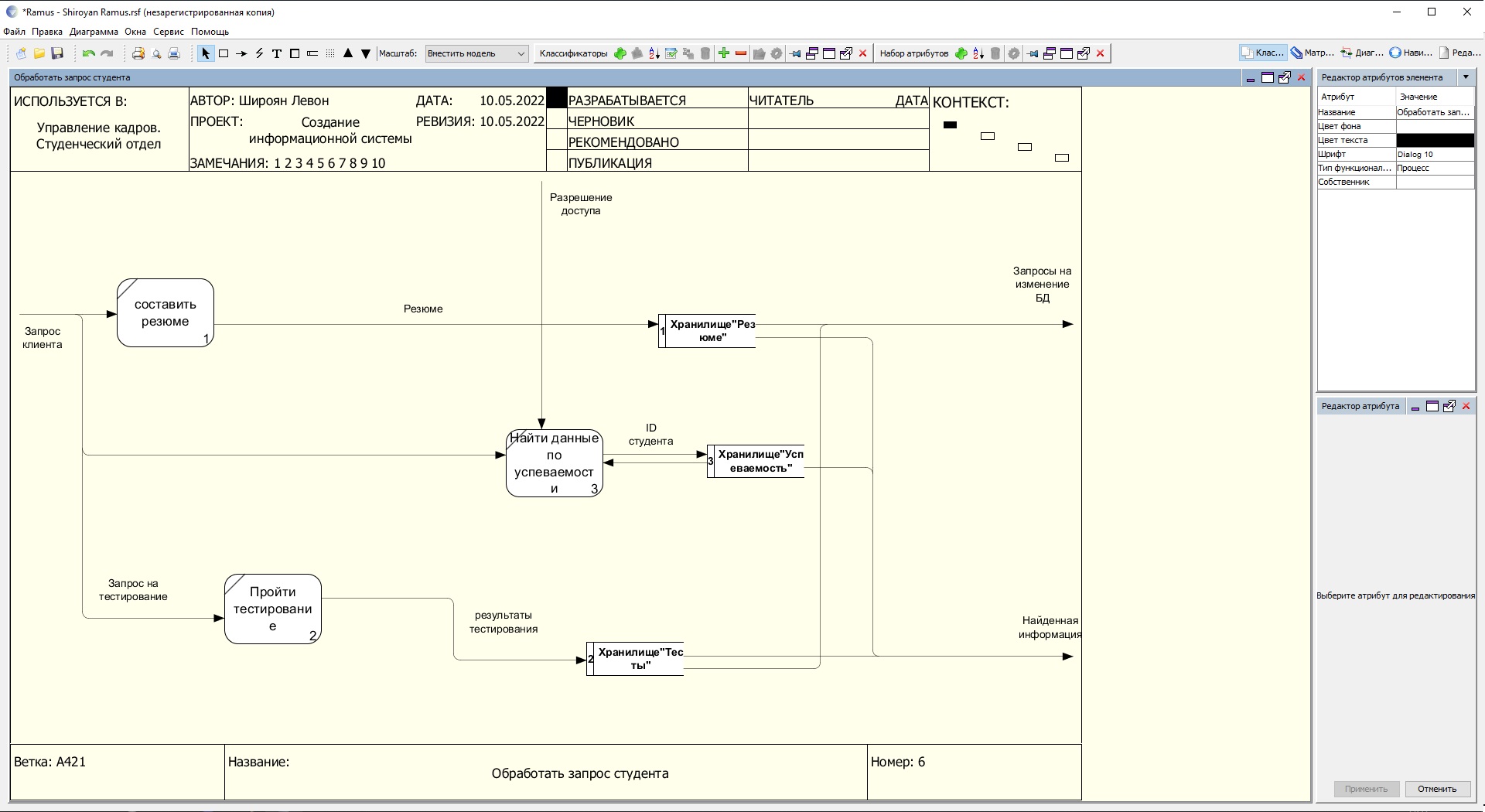


Рисунок 24. Пример создания функциональной модели DFD

Теперь необходимо создать классификаторы:

1. Хранилище «Резюме»
2. Хранилище «Тесты»
3. Хранилище «Успеваемость»

Для создания классификаторов переключиться на соответствующее окно кнопкой находящейся на верхней инструментальной панели и в появившемся окне Классификаторы (вкладка Классификаторы) с помощью кнопки создать необходимые классификаторы.

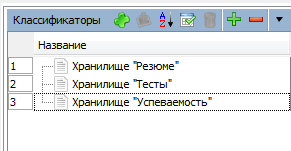
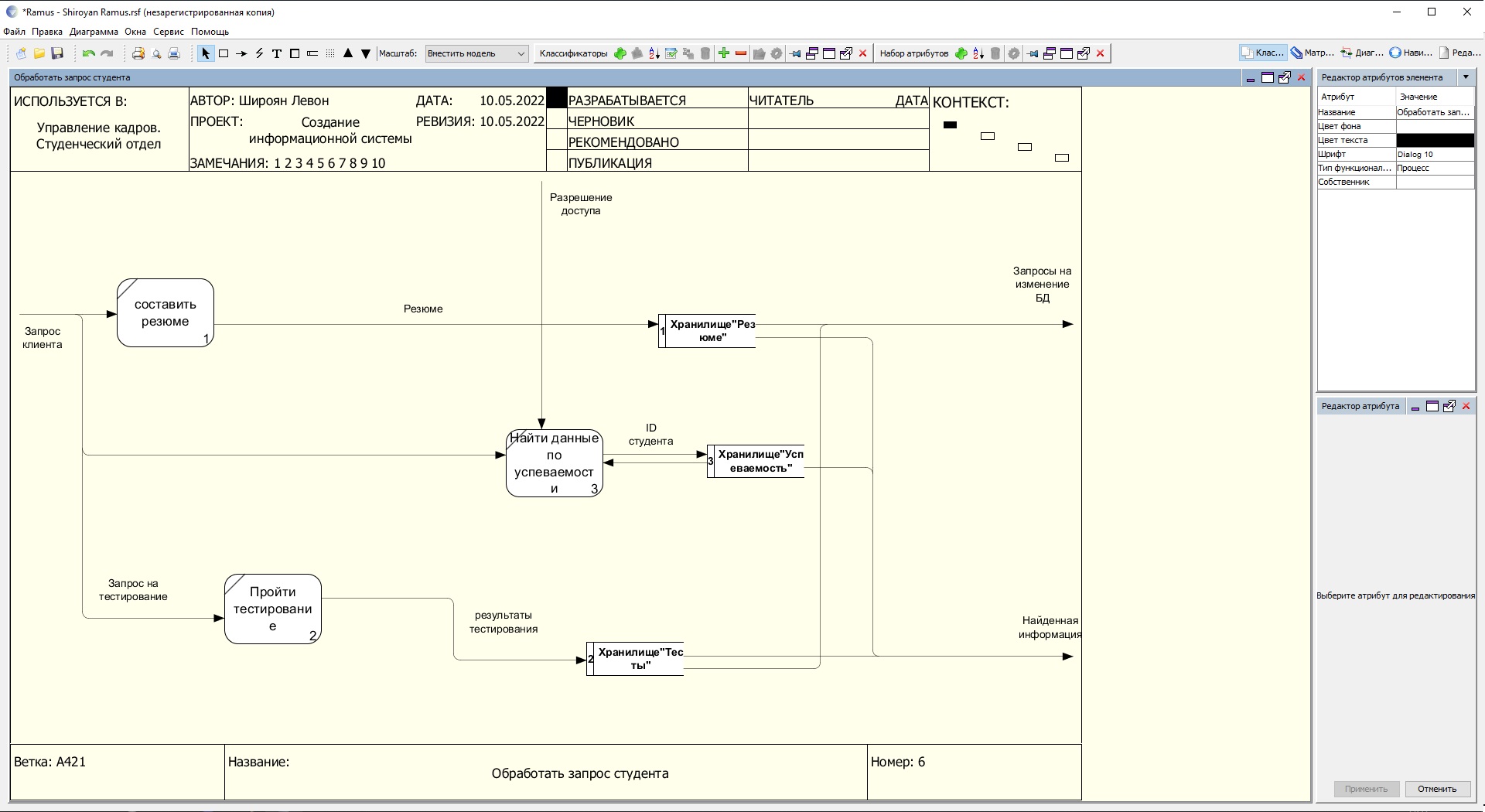


Рисунок 25. Классификаторы

1. Внесите в модель соответствующие хранилища данных при помощи кнопки .
2. И постройте DFD модель процесса.

Рисунок 26. Пример создания функциональной модели DFD



Дополните харнилище элементом: Хранилище «Экспертные оценки»

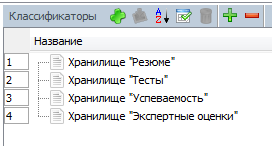


Рисунок 27.Классификаторы

Постройте DFD модель процесса «Обработать запрос эксперта»

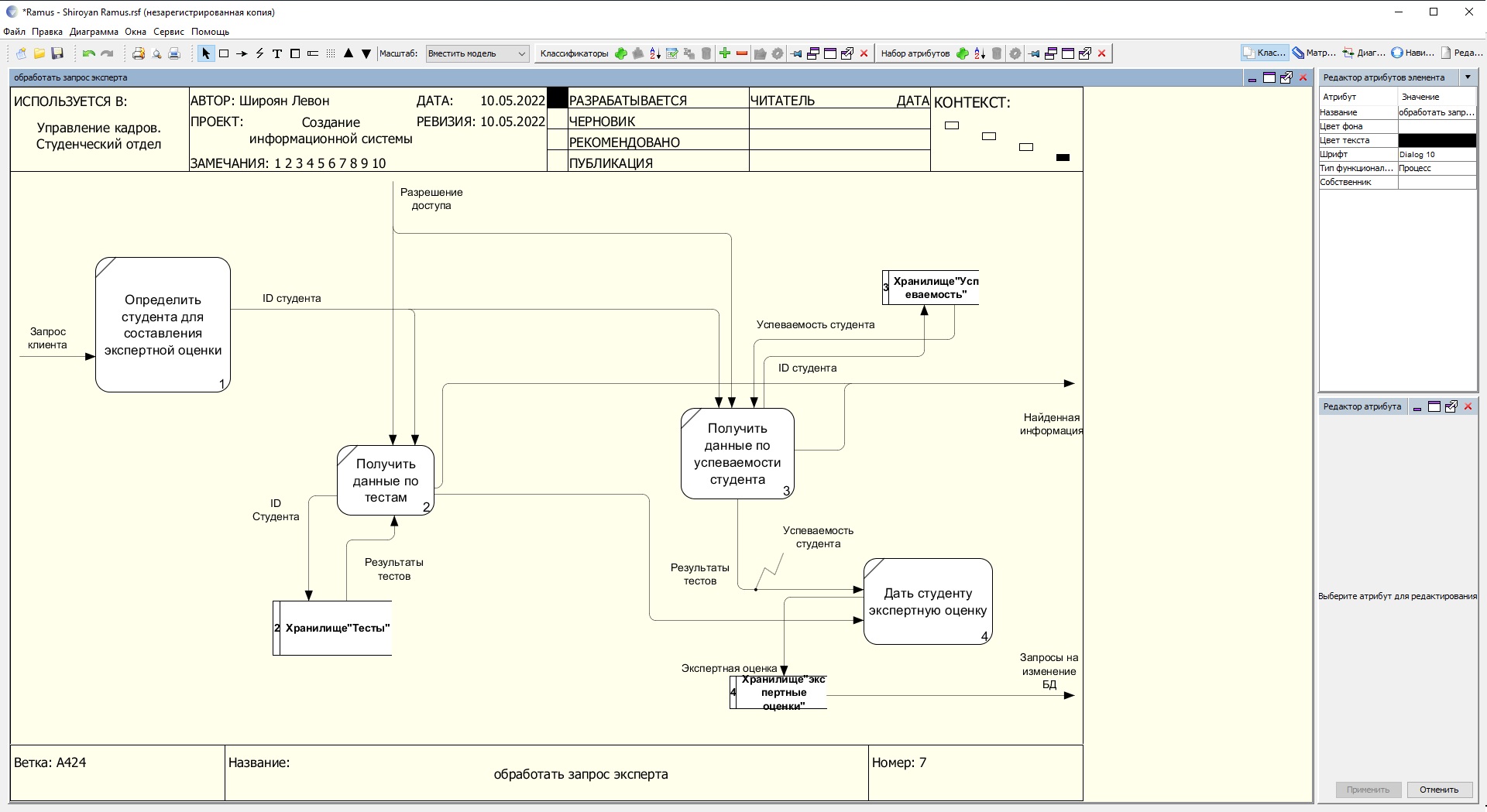


Рисунок 28. Пример создания функциональной модели DFD

**ЗАКЛЮЧЕНИЕ**

Итак, мы выполнили функционально-структурный анализ предметной области «Работа библиотеки». Наши представления о предметной области мы систематизировали в виде ее функциональной модели. Функциональная модель разрабатывалась на основе методологии IDEF0. В результате выполнения конкретной работы мы познакомились с методологией функционального моделирования и технологией работы в среде ее реализации Ramus Educational.

Важно понять, что IDEF0 представляет собой универсальную методологию анализа бизнес-процессов, а, следовательно, она может быть полезна для специалистов в любой области профессиональной деятельности. Методология IDEF0 предоставляет возможность заинтересованному профессионалу выполнить критический анализ действительного состояния профессиональной деятельности и попытаться выполнить ее реструктуризацию для устранения обнаруженных недостатков.

При выполнении функционально-структурного анализа по методологии IDEF0 большое значение имеют компьютерные средства его реализации. В настоящее время известно множество CASE-систем такого назначения. Знакомство с Ramus Educational позволит начинающему пользователю составить представление о технологии работы в таких системах.

Если начинающему разработчику функциональных моделей понятны концепции методологии IDEF0 и он осознал как ее применять, если он освоил технологию разработки модели в CASE-системе, например Ramus Educational, то можно считать что главная цель учебного издания достигнута.

**БИБЛИОГРАФИЧЕСКИЕ ИСТОЧНИКИ**

1. PDF файл Grekoul
2. [DFD: примеры и правила построения диаграмм потоков данных (bpmn.pro)](https://bpmn.pro/process/dfd?ysclid=l37mxwu2og) Главы 1 и 2
3. [Стандарт idef0 примеры. IDEF0 диаграмма: примеры и правила построения (technolakpiter.ru)](https://technolakpiter.ru/standart-idef0-primery-idef0-diagramma-primery-i-pravila-postroeniya/) Глава 1
4. 5.[Вступление (https://spravochnick.ru/informatika/ramus\_educational\_opisanie\_i\_sfery\_primeneniya/)](http://www.kgau.ru/sveden/2017/ieu/metod_090303_23.pdf) Вступление
5. [Что обозначает в idef0 понятие декомпозиция. Принципы построения модели idef0. Принципы ограничения сложности IDEF0-диаграмм (gostehstroy.ru)](https://gostehstroy.ru/niche-selection/chto-oboznachaet-v-idef0-ponyatie-dekompoziciya-principy/) Глава1
6. Ramus методичка Финансы и кредит Решение задач Экономика и финансы Задача 1 - Финансы и кредит Решение задач Экономика и финансы (https://www.studocu.com/ru/document/finansovyy-universitet-pri-pravitel%D1%8Cstve-rf/ekonomika/ramus-metodichka-finansy-i-kredit-reshenie-zadach-ekonomika-i-finansy-zadacha-1-finansy-i-kredit-reshenie-zadach-ekonomika-i-finansy/21839228)

ПРИЛОЖЕНИЕ 1

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ МОСКОВСКОЙ ОБЛАСТИ**

**«НОГИНСКИЙ КОЛЛЕДЖ»**

Д Н Е В Н И К

**учета учебных работ**

**при обучении**

**Ф.И.О. учащегося Широяна Левона Багратовича**

**Группа № 2ИСПр**

**Профессия (специальность)** **Информационные системы и программирование**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата | № и наименование темы программы | Содержание выполненных работ | Оценка | Подпись ответственного лица |
| 12.04.2022 | Анализ предметной области | Изучение теоретических материалов |  |  |
| 12.04.2022 | Анализ предметной области | Изучение теоретических материалов |  |  |
| 12.04.2022 | Анализ предметной области | Изучение теоретических материалов |  |  |
| 12.04.2022 | Анализ предметной области | Изучение теоретических материалов методологии idef0 |  |  |
| 12.04.2022 | Анализ предметной области | Изучение теоретических материалов методологии idef0 |  |  |
| 12.04.2022 | Анализ предметной области | Изучение теоретических материалов методологии idef0 |  |  |
| 18.04.2022 | Описание бизнес-процессов предметной области | Поиск материалов для создания контекстной диаграммы IDEF0 |  |  |
| 18.04.2022 | Описание бизнес-процессов предметной области | Поиск материалов для создания контекстной диаграммы IDEF0 |  |  |
| 18.04.2022 | Описание бизнес-процессов предметной области | Поиск материалов для создания контекстной диаграммы IDEF0 |  |  |
| 18.04.2022 | Описание бизнес-процессов предметной области | Поиск материалов для создания контекстной диаграммы IDEF0 |  |  |
| 18.04.2022 | Описание бизнес-процессов предметной области | Поиск материалов для создания контекстной диаграммы IDEF0 |  |  |
| 18.04.2022 | Описание бизнес-процессов предметной области | Поиск материалов для создания контекстной диаграммы IDEF0 |  |  |
| 25.04.2022 | Сбор данных для создания информационной системы | Создание контекстной диаграммы IDEF0 |  |  |
| 25.04.2022 | Сбор данных для создания информационной системы | Создание контекстной диаграммы IDEF0 |  |  |
| 25.04.2022 | Сбор данных для создания информационной системы | Создание контекстной диаграммы IDEF0 |  |  |
| 25.04.2022 | Сбор данных для создания информационной системы | Создание контекстной диаграммы IDEF0 |  |  |
| 25.04.2022 | Сбор данных для создания информационной системы | Создание контекстной диаграммы IDEF0 |  |  |
| 25.04.2022 | Сбор данных для создания информационной системы | Изучение теоретических материалов методологии DFD |  |  |
| 03.05.2022 | Создание информационной системы в программе Ramus | Изучение теоретических материалов методологии DFD |  |  |
| 03.05.2022 | Создание информационной системы в программе Ramus | Изучение теоретических материалов методологии DFD |  |  |
| 03.05.2022 | Создание информационной системы в программе Ramus | Изучение теоретических материалов методологии DFD |  |  |
| 03.05.2022 | Создание информационной системы в программе Ramus | Поиск материалов для создания контекстной диаграммы DFD |  |  |
| 03.05.2022 | Создание информационной системы в программе Ramus | Поиск материалов для создания контекстной диаграммы DFD |  |  |
| 03.05.2022 | Создание информационной системы в программе Ramus | Поиск материалов для создания контекстной диаграммы DFD |  |  |
| 16.05.2022 | Создание информационной системы в программе Ramus | Поиск материалов для создания контекстной диаграммы DFD |  |  |
| 16.05.2022 | Создание информационной системы в программе Ramus | Создание контекстной диаграммы DFD |  |  |
| 16.05.2022 | Создание информационной системы в программе Ramus | Создание контекстной диаграммы DFD |  |  |
| 16.05.2022 | Создание информационной системы в программе Ramus | Создание контекстной диаграммы DFD |  |  |
| 16.05.2022 | Создание информационной системы в программе Ramus | Создание контекстной диаграммы DFD |  |  |
| 16.05.2022 | Создание информационной системы в программе Ramus | Создание контекстной диаграммы DFD |  |  |
| 18.05.2022 | Оформление отчета | Создание отчета |  |  |
| 18.05.2022 | Оформление отчета | Создание отчета |  |  |
| 18.05.2022 | Оформление отчета | Создание отчета |  |  |
| 18.05.2022 | Оформление отчета | Создание отчета |  |  |
| 18.05.2022 | Оформление отчета | Создание отчета |  |  |
| 18.05.2022 | Оформление отчета | Создание отчета |  |  |

Приложение 2

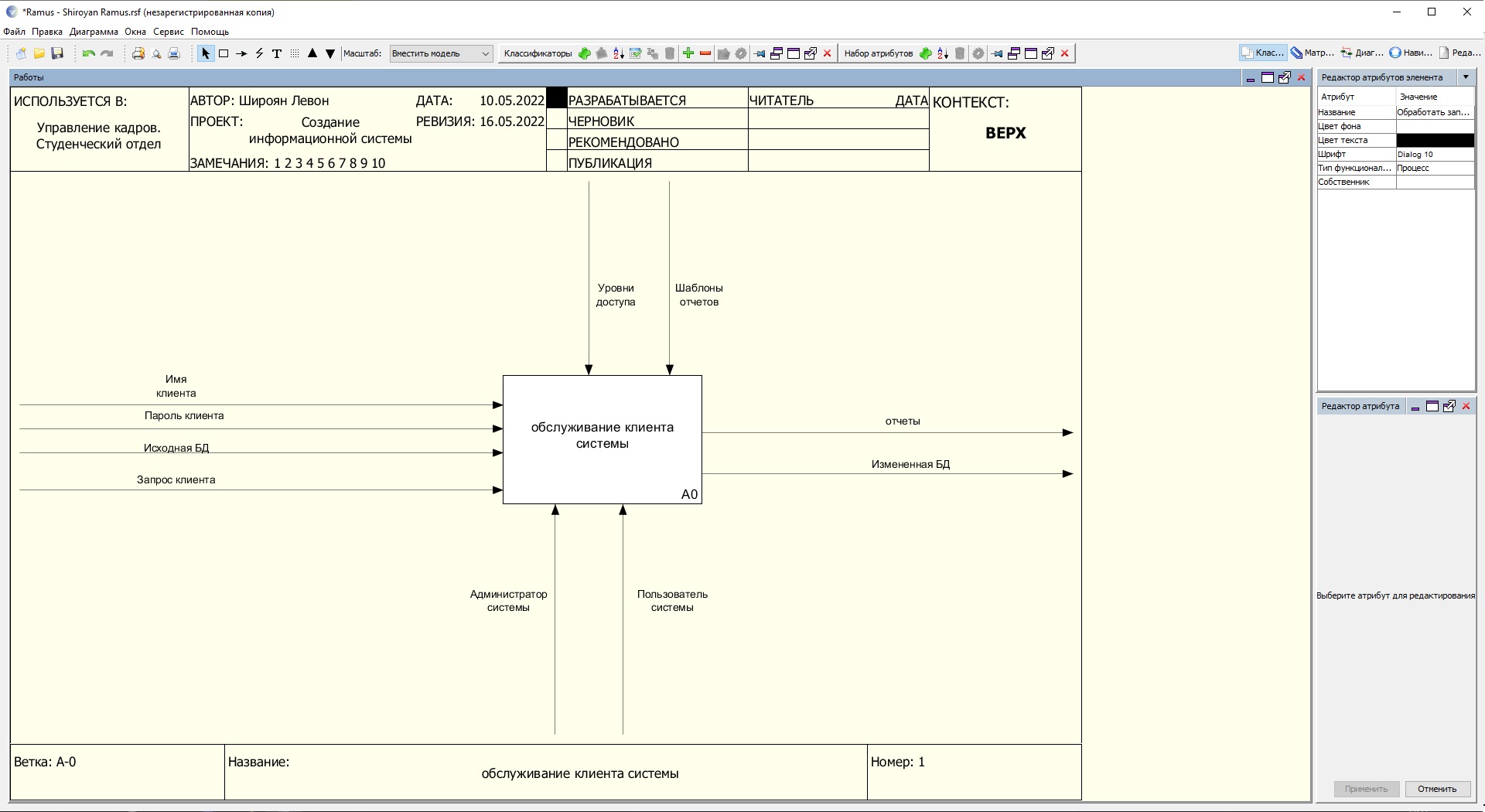


Рисунок 1.Главная страница

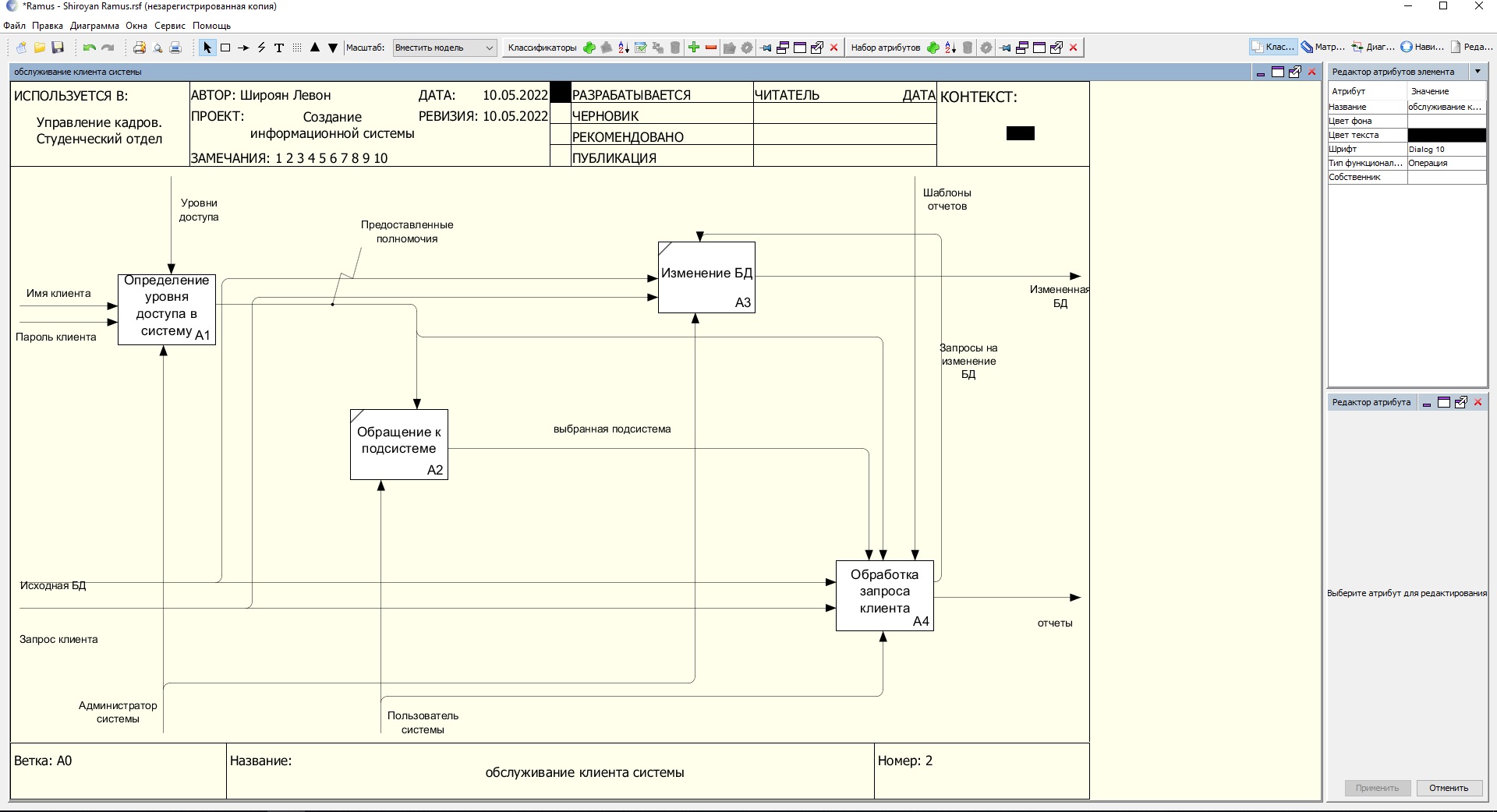


Рисунок 2.Обслуживание клиента системы

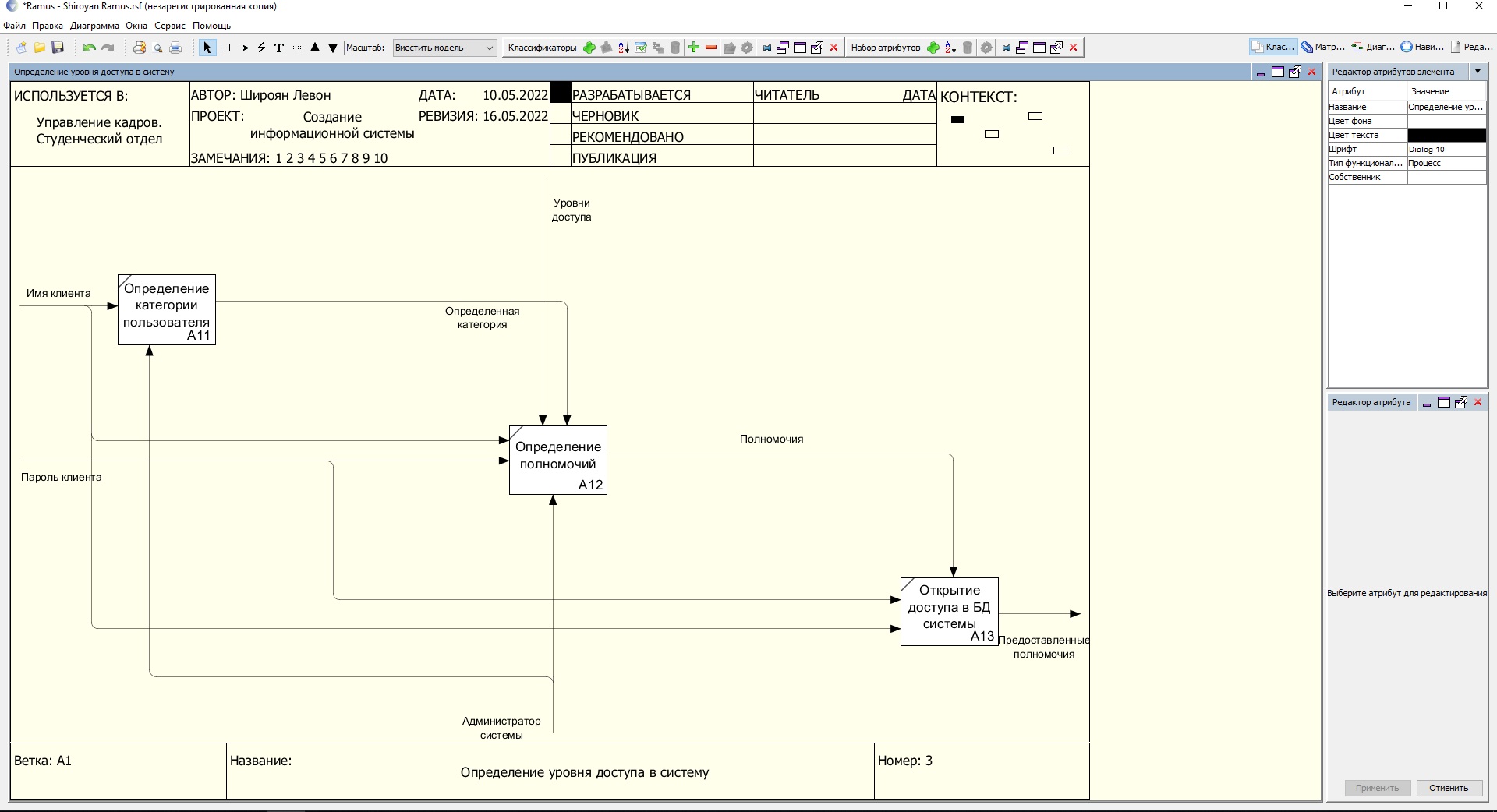


Рисунок 3. Определение уровня доступа в систему

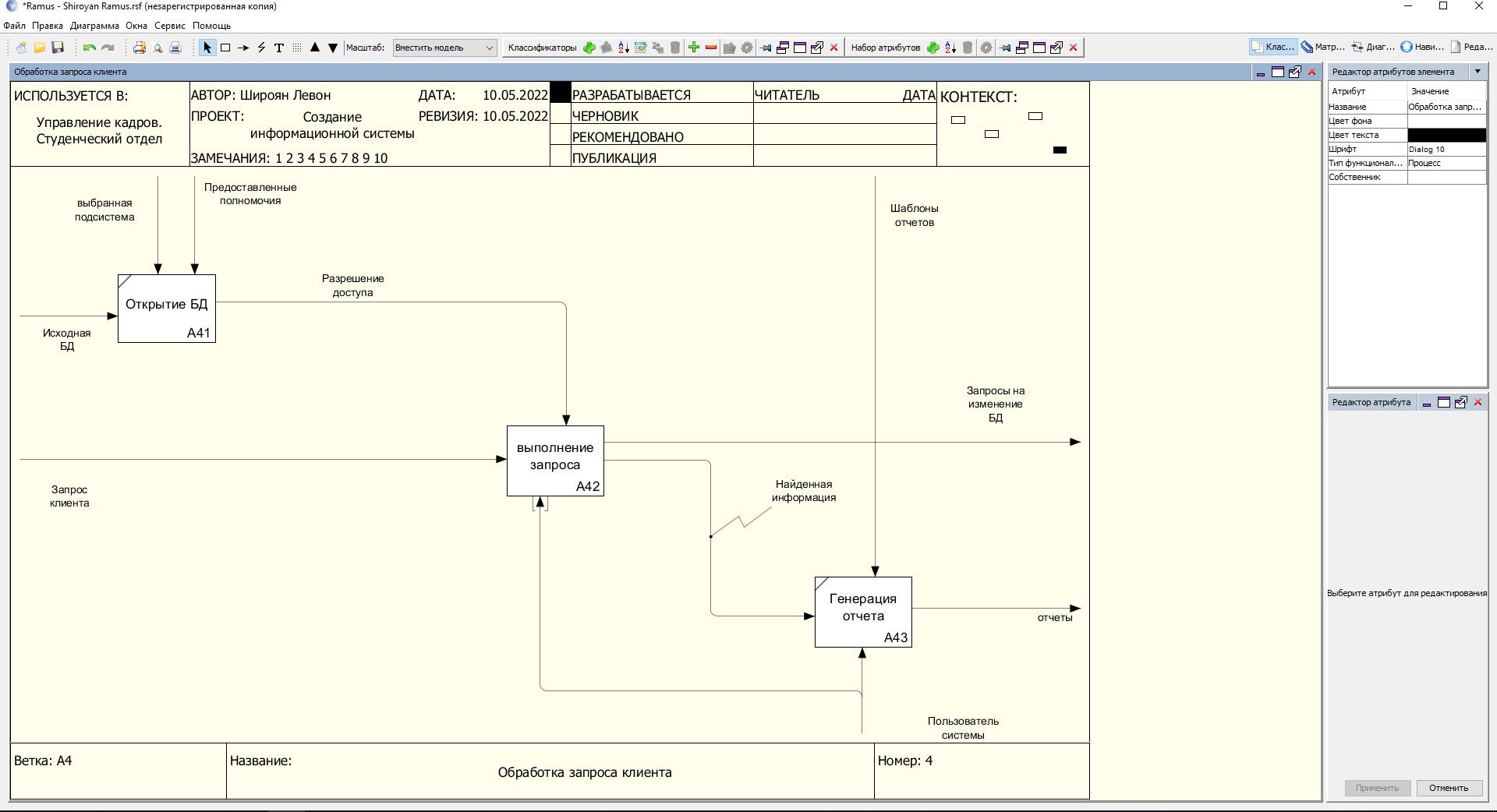


Рисунок 4. Обработка запроса клиента

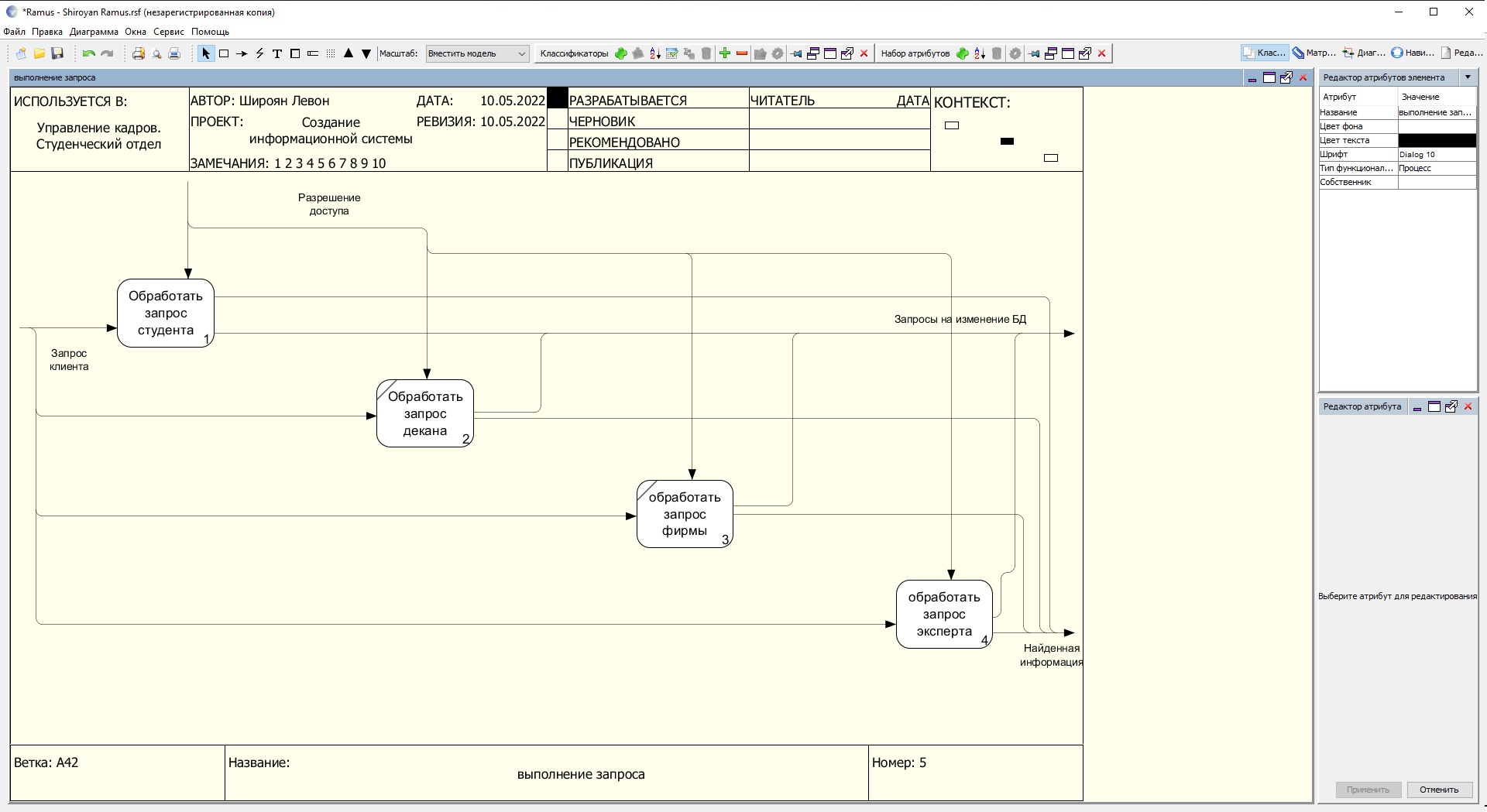


Рисунок 5. Выполнение запроса

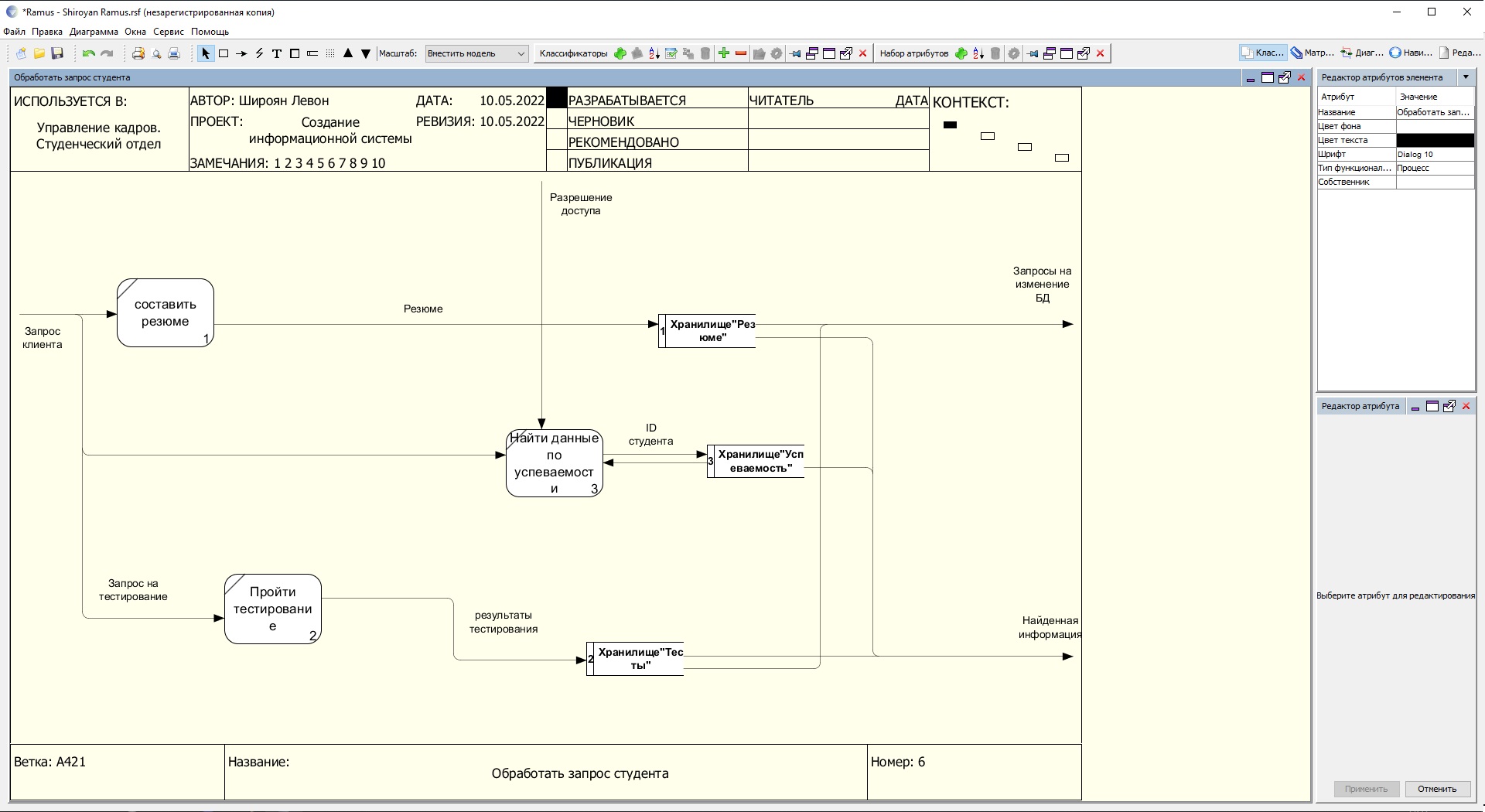


Рисунок 6. Обработать запрос эксперта

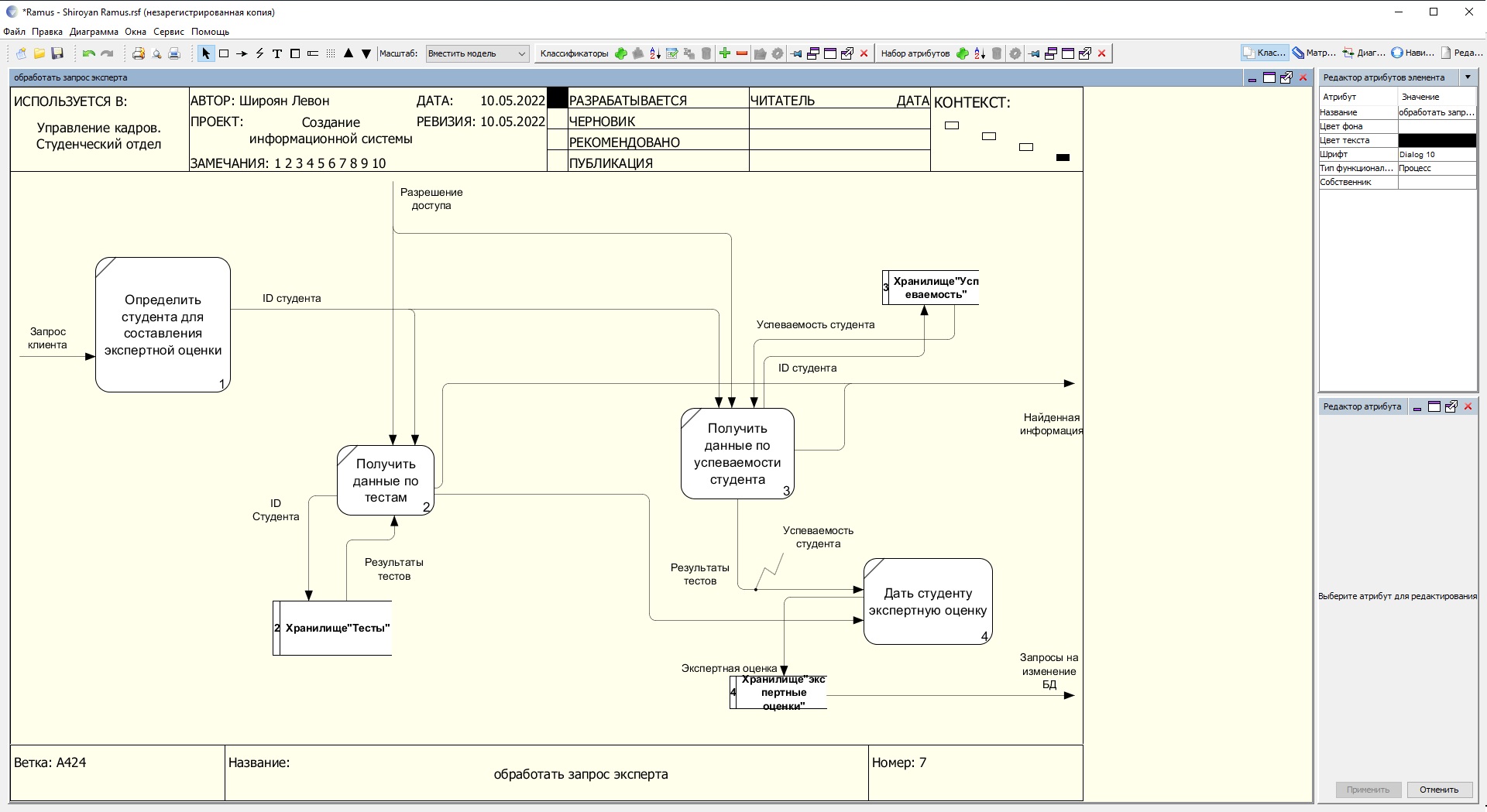


Рисунок 7. Обработать запрос студента.