Содержание

[Введение 3](#_Toc89710890)

[1 Техническое задание 4](#_Toc89710891)

[1.1 Анализ предметной области 4](#_Toc89710892)

[1.2 Постановка задачи 5](#_Toc89710893)

[2 Технический проект информационной системы 8](#_Toc89710894)

[2.1 Функциональная модель 8](#_Toc89710895)

[2.1.1 Контекстная диаграмма и диаграммы детализации процессов 8](#_Toc89710896)

[2.1.2 Диаграмма дерева узлов 1](#_Toc89710897)2

[2.2 Информационная модель 1](#_Toc89710898)3

[2.2.1 Идентификация сущностей и связей. ER-диаграмма логического уровня….. 1](#_Toc89710899)3

[2.2.2 ER-диаграмма физического уровня. Ограничения ссылочной целостности. Переопределение триггеров. Индексирование отношений 1](#_Toc89710900)6

[2.2.3 Определение представлений, хранимых процедур серверной компоненты, ER-диаграмма в режиме отображения представлений 1](#_Toc89710901)7

[2.3 Верификация спроектированной логической модели 1](#_Toc89710902)7

[3 Реализация системы 2](#_Toc89710903)0

[3.1 T-SQL-определения регламентированных запросов 2](#_Toc89710904)0

[3.2 T-SQL-определения триггеров 2](#_Toc89710905)3

[3.3 T-SQL-определения хранимых процедур 2](#_Toc89710906)5

[3.4 T-SQL-определения курсоров 3](#_Toc89710907)3

[3.5 Описание клиентских приложений 3](#_Toc89710908)5

[4 Результаты тестирования информационной системы 4](#_Toc89710909)1

[Заключение 4](#_Toc89710910)2

[Список литературных источников 4](#_Toc89710911)3

# Введение

Обсерватория солнечной динамики, (англ. Solar Dynamics Observatory, SDO) — космическая обсерватория НАСА для изучения Солнца, рассчитанная на 5 лет работы. Была запущена 11 февраля 2010 года в рамках программы «Жизнь со Звездой» (Living With a Star, LWS). Целью программы LWS является развитие научных знаний, необходимых для эффективного решения аспектов Солнечно-Земных связей, которые непосредственно влияют на жизнь и общество. Целью SDO является понимание влияния Солнца на Землю и околоземное пространство путём изучения солнечной атмосферы на малых масштабах времени и пространства и во многих длинах волн единовременно.

Создаваемая информационная система предназначена для автоматизации деятельности с целью повышения эффективности сбора информации, упрощения занесения статистической информации и создания отчетов, а также для оптимизации деятельности сотрудников в целом для создания учетных ведомостей.

Разработанная информационная система позволит значительно упростить процесс заполнения и создания отчетных документов, при этом устранив количество совершаемых в силу человеческого фактора ошибок и значительно ускоряя работу.

# Техническое задание

# Анализ предметной области

Темой данного курсового проекта является автоматизация работы Обсерватории. В информационной области значение оптимизации нельзя недооценить: от этого может зависеть финансовое состояние.

Актуальность изучаемого вопроса безусловна: автоматизация – верный путь в повышении эффективности. Автоматизация позволяет сократить время выполнения рутинных механических действий, избежать множества ошибок, допускаемых человеком, тем самым повысить эффективность труда, оптимизировать деятельность, освободить время для решения более важных задач и выйти на новый уровень.

Объектом исследования в данной курсовой работе является Обсерватория солнечной динамики.

Создаваемая информационная система предназначена для автоматизации деятельности отделов Обсерватории с целью повышения эффективности наблюдения за космическими объектами, созданием ежедневных отчетах, взаимодействию с клиентами, использваему оборудованию и его состоянию, сотрудниках Обсерватории.

В базе будут введены следующие категории пользователей: модератор, клиент 1, клиент 2, государственный клиент, пользователь.

Модератору будут доступны все возможности сайта для полного контроля над всеми событиями. Клиенту 1 будут доступны возможности взаимодействия лишь для продажи своего товара. Клиент 2 поддержке будет доступен так же весь функционал для устранения возможных багов на сайте. Пользователю доступна лишь возможность куплю и просмотра товаров на сайте.

Обсерватория включает следующие рабочие места:

* Космонавт.
* Инженер-конструктор.
* Инженер-программист.
* Астроном.
* Космический баллистик.
* Водитель.
* Космический биолог.
* Инженер-конструктор.

Каждое рабочее место интернет магазина выполняет свои функции.

Данная ИС позволит значительно упростить процесс создания отчетных документов о товарах, при этом устранив количество совершаемых в силу человеческого фактора ошибок и значительно ускоряя работу.

# Постановка задачи

Система должна удовлетворять следующим требованиям.

Практичность (Usability): данная система должна быть легкой в освоении. Квалифицированный персонал: индивидуальные предпринимали, покупатели и продавцы – будут в состоянии разобраться с принципом работы с разрабатываемой ИС за минимальное время и выполнять свои обычные текущие задачи более продуктивным образом.

Надежность системы (Reliability): Система должна быть доступна для операционного использования в течение указанного времени: 99%. Среднее время восстановления должно принимать среднее значение: 90% всех системных сбоев должны ликвидироваться за 5 минут, а 99.9% — в течение часа. Особенных требований к точности нет.

Разработанная система должна предоставлять возможность ввода, редактирования, удаления информации и предоставлять возможность группировать имеющиеся данные по определенным критериям.

Безопасность и защита данных: будет предоставлен полный доступ для Администраторов и технической поддержке, когда как частичный доступ будет осуществляться для посетителей и клиентов.

Функции, выполняемые информационной системой:

1. Выдача информации о космических объектах. Без использования автоматизированной системы выдача информации о космических объектах будет занимать значительное время. Вся информация храниться в одной базе данных, доступ к которой осуществляется по названию космического объекта и его индикационному номеру.
2. Поиск сотрудника по стажу работы. Раньше для поиска сотрудника приходилось получать его дату начало контракта, что затрудняло вычисление стажа работы из-за большого количества персонала.
3. Выдача информации о клиентах обсерватории. Каждый клиент может ознакомиться со списком постоянных клиентов и их сфере продвижения проекта, что помогает новым клиентам лучше адаптироваться и понимать в каком направлении им принимать участие в спонсорстве.
4. Обновление расположения космических объектов в пределах галактики. Расположение объектов фиксировалось только с одного ракурса по постоянной частоте обновления. Сейчас космический объект фотографирует из разных точек космоса и земли. После чего мы получаем точное расположение любого космического объекта.
5. Сортировка космических объектов по их размерам. До этого существовала сортировка по названию объекта. Сейчас же можно отсортировать по дате обнаружения, скорости полета, опасности, яркости, что облегчает поиск по их критериям.
6. Изменение языка системы на русский и английский. Использование одного языка системы затрудняет использование ее людьми, которые знакомы только с родным языком. По этому введение изменения языка текста системы на другие иностранные языки приведет к пониманию предназначения данной системы.
7. Возможность отправлять посетителей обсерватории в черный список. Система будет автоматически отправлять неугодных посетителей, или клиентов, или сотрудников в черный список спустя определенное истечение времени и количества жалоб.
8. Формирование ежедневных отчетов для обнаружении небесных тел. До этого момента ежедневными отчетами занимались группа людей, из-за большого потока новой информации о новых небесных телах, принято решение передавать все в общую базу данных о новых небесных телах. От туда брать информацию и составлять отчет по пройденному дню.
9. Вывод способа получения денежных средств. Существовал общий внутренний корпоративный счет, который обрабатывал все транзакции всех клиентов, посетителей, работников. Сейчас же каждый может выбрать свой способ получения денежных средств с помощью любого банка, выдачи чека, в виде акций компании.
10. Получение информации о сувенирном магазине. Раньше посетители могли узнать о новых сувенирах, только если находились в самом магазине. Сейчас же создана специальная база данных для информации о сувенирном магазине, который обновляется при новом поступлении товара. Поэтому узнать о необходимом товаре стало легче.
11. EVE(Extreme Ultraviolet Variability Experiment). Использовались тепловые, инфракрасные излучения и видимый свет. EVE будет измерять солнечное излучение ультрафиолетовых волнах – примерно каждые 10 секунд , что позволит нам использовать это дял предупреждения людей об опасностях вспышек на солнце, описывать модели движения, которые рассказывают нам, что происходит с атмосферой, так же позволить людям, которые занимаются радиосвязью и навигацией, знать, что в их системе может быть проблема –з-за вспышке на луне.
12. Сборка визуализации атмосферы. До этого изображения солнца передавались на 10-ти различных длинах волн каждые 1.8 часа. Поэтому ученым, что бы собрать около 100000 изображений на всех длинах волн понадобиться около 2 года. С применением длин волн близких к ультрафиолетовым, изображения получается за 12 секунд.
13. Гелиосейсмический и магнитный имидж-сканер. Раньше сканирование состояния магнитных полей солнца затрудняло из-за радио помех генерируемых солнцем. Новый магнитный имидж-сканер устойчив к изменениям магнитной активности, что позволяет понять солнечную изменчивость с минимальными ошибками в расчетах.
14. Связь. В нормальном режиме работы космический аппарат передает непрерывный поток данных со скоростью 200МБ/с с фотографиями и другими изменениями через NASA DEEP Space Network наземных станций. Сейчас с новейшими технологиями радио антенн скорость передачи данных составляет до 10ГБ/с, что обеспечивает более качественное изображение и точные координаты.
15. Добавление частых посетителей в базу «постоянные клиенты». Раньше была одна общая база со всеми клиента, где не учитывалось количество посещений, участие в новых проектах. Сейчас же для каждого клиента вводят новый атрибут – это активность в жизни Обсерватории. Чем выше этот показатель, тем с большей вероятностью он попадет в список «постоянные клиенты».

# Технический проект информационной системы

# Функциональная модель

# Контекстная диаграмма и диаграммы детализации процессов

Контекстная диаграмма и диаграмма детализации процессов представлена на рисунке 1.

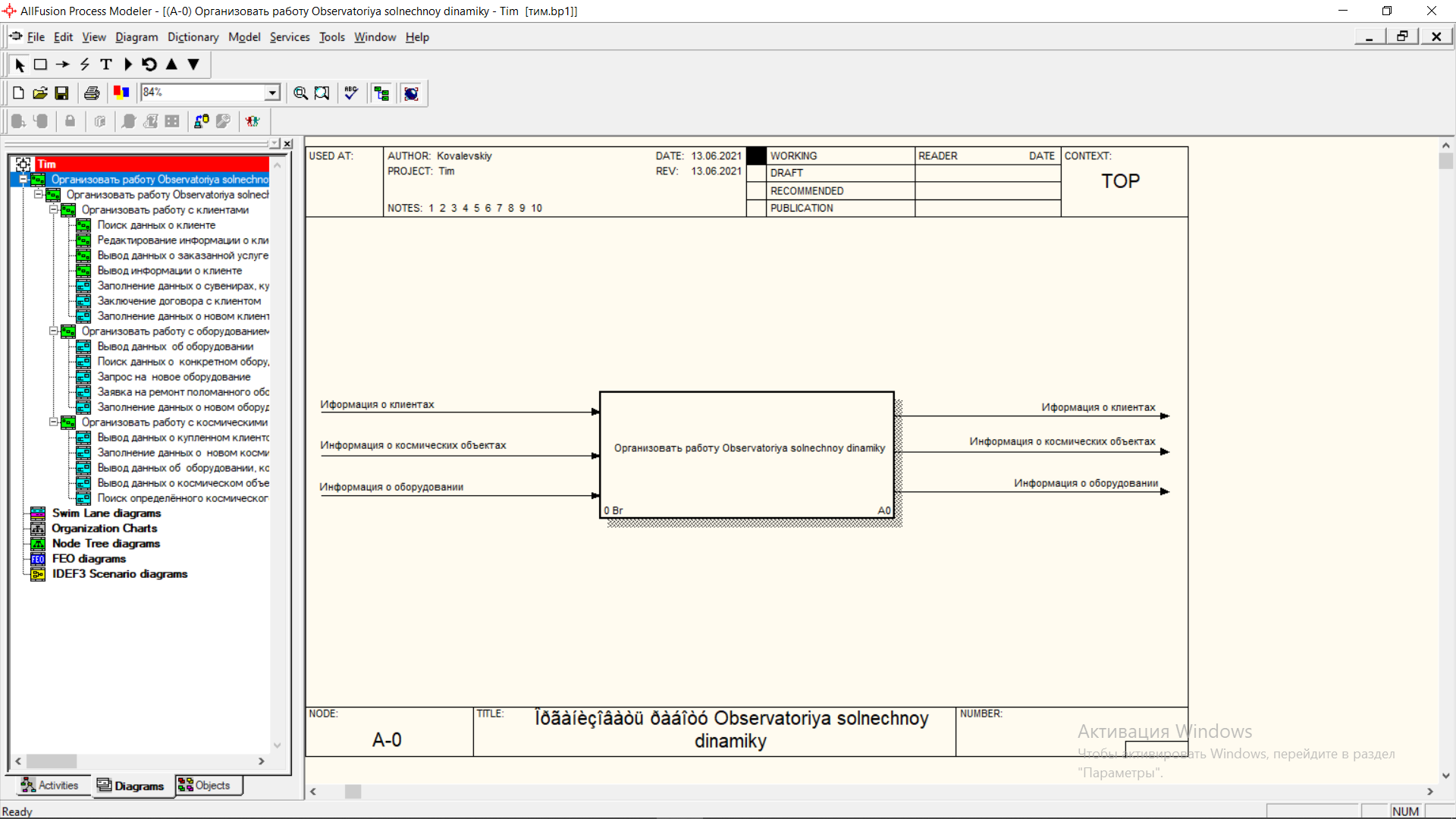


Рисунок 1 – Контекстная диаграмма

Вся диаграмма разбита на уровни, где верхним (А0) является процесс «Организовать работу Обсерватории солнечной динамики». Далее уровни будут подвергаться декомпозиции для более наглядной картины функционирования системы.

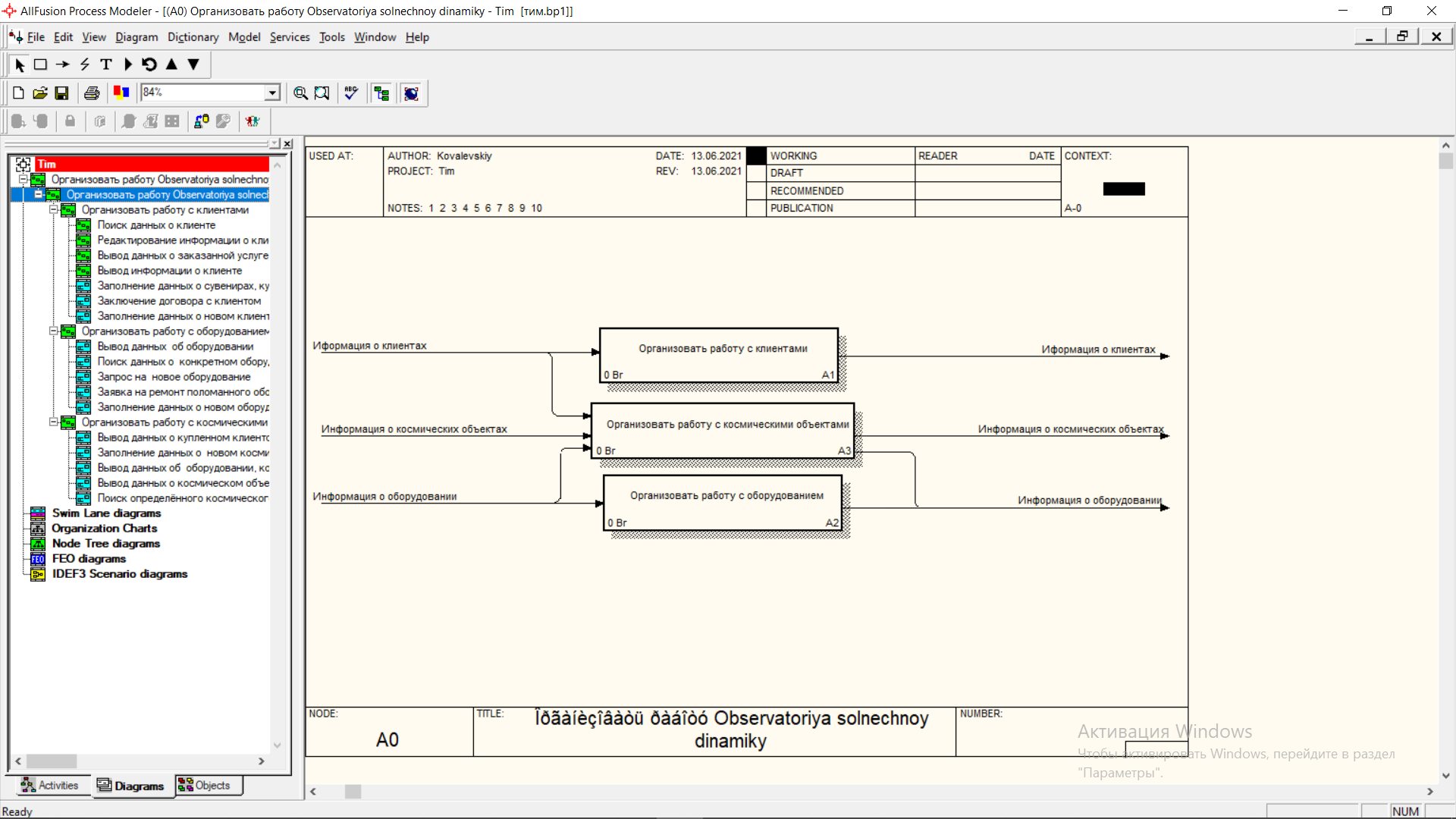


Рисунок 2 – Декомпозиция процесса А0 «Организовать работу Обсерватории солнечной динамики»

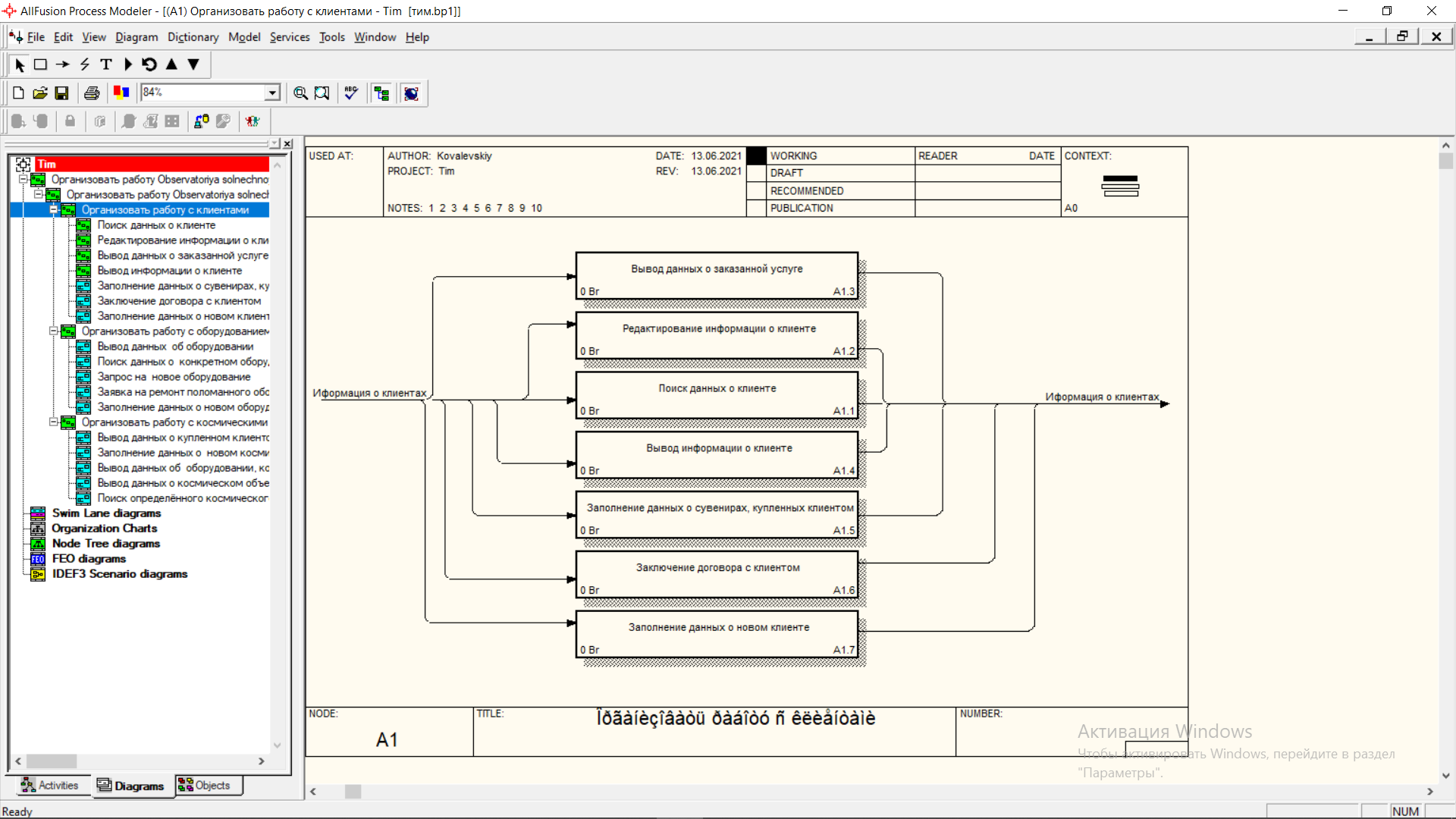


Рисунок 3 – декомпозиция процесса «Организовать работу с клиентами»

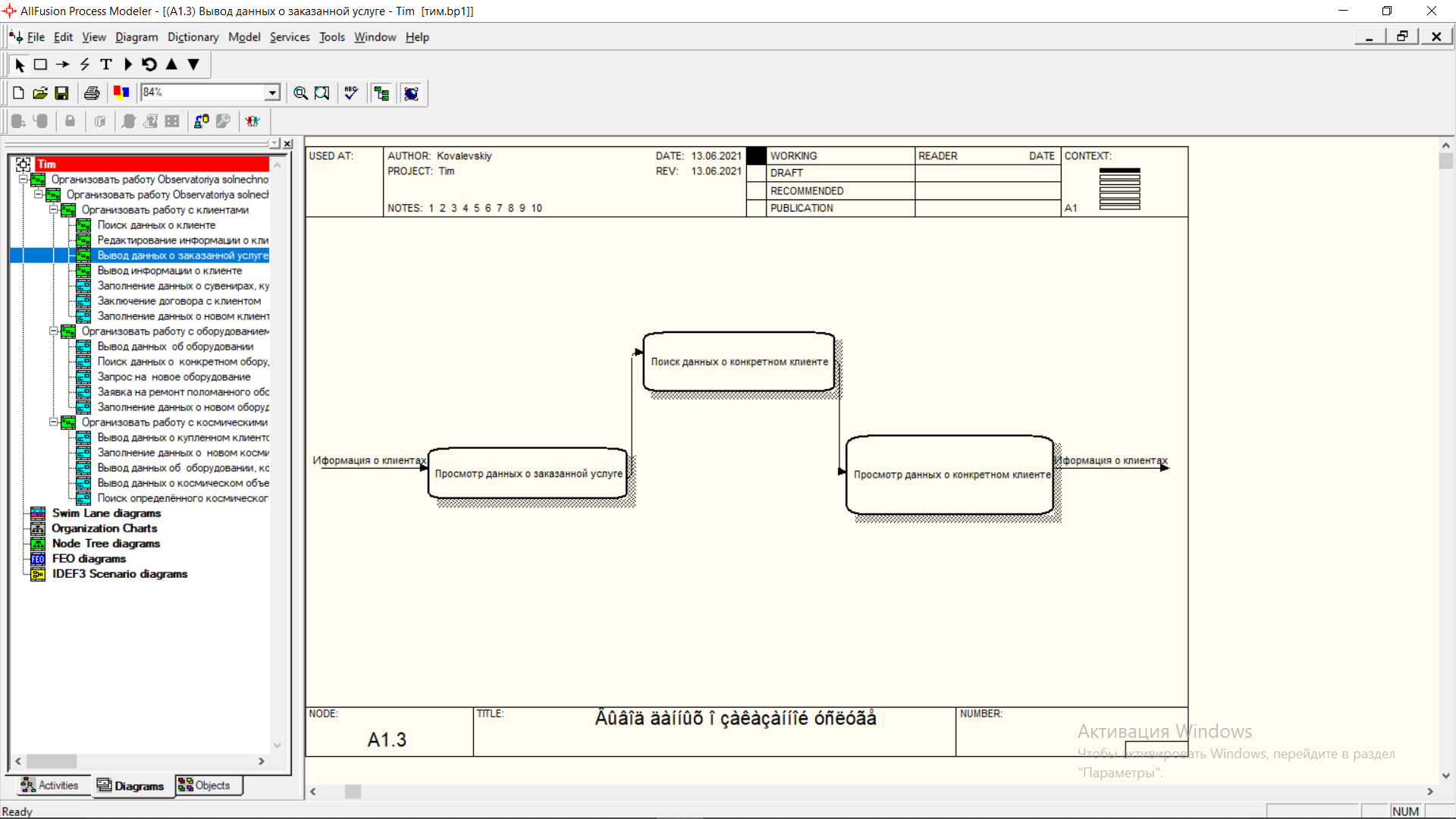


Рисунок 4 – декомпозиция процесса «Вывод данных о заказанной услуге»

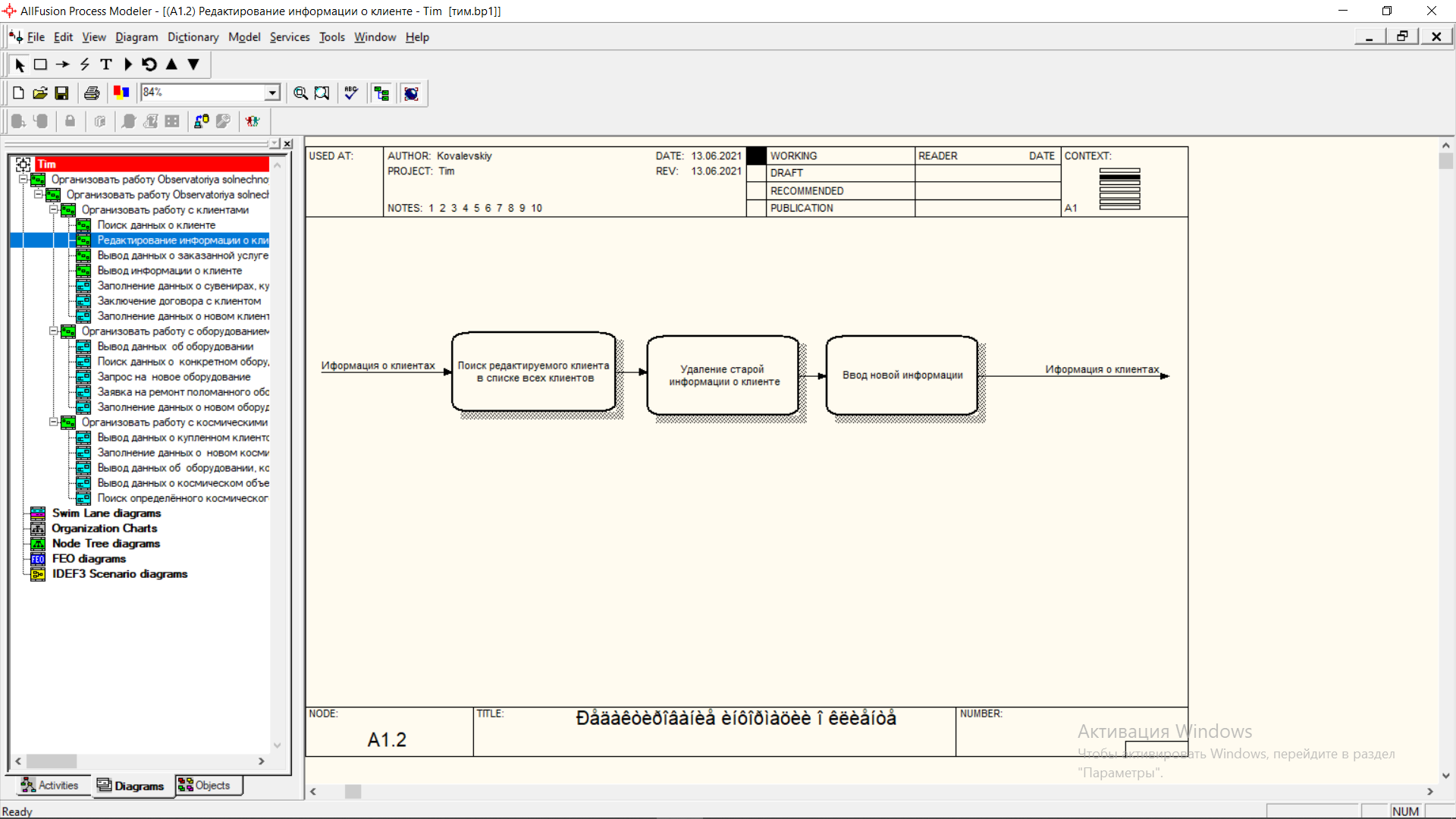


Рисунок 5 – декомпозиция процесса «Редактирование информации о клиенте»

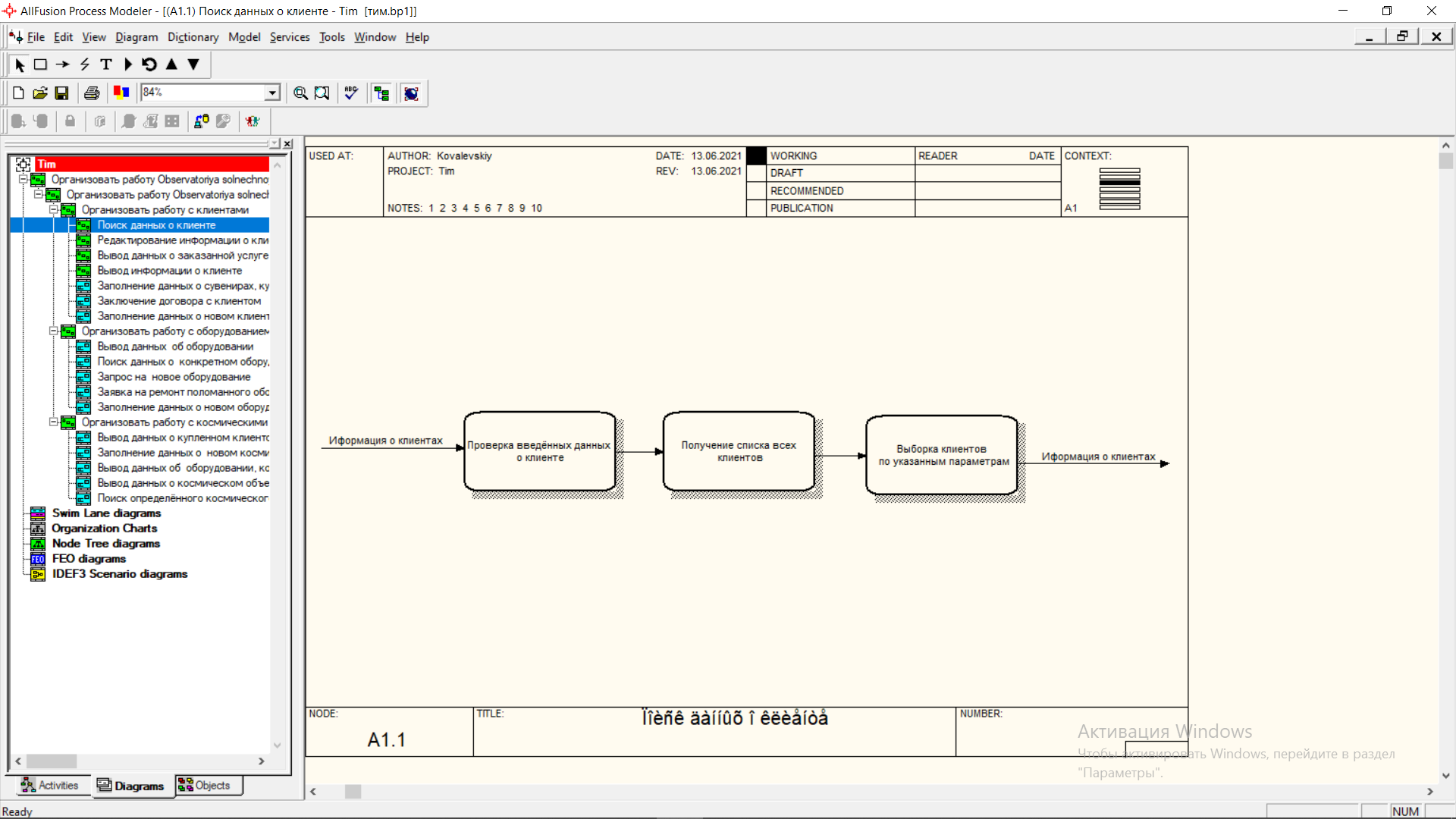


Рисунок 6 – декомпозиция процесса «Поиск данных о клиенте»

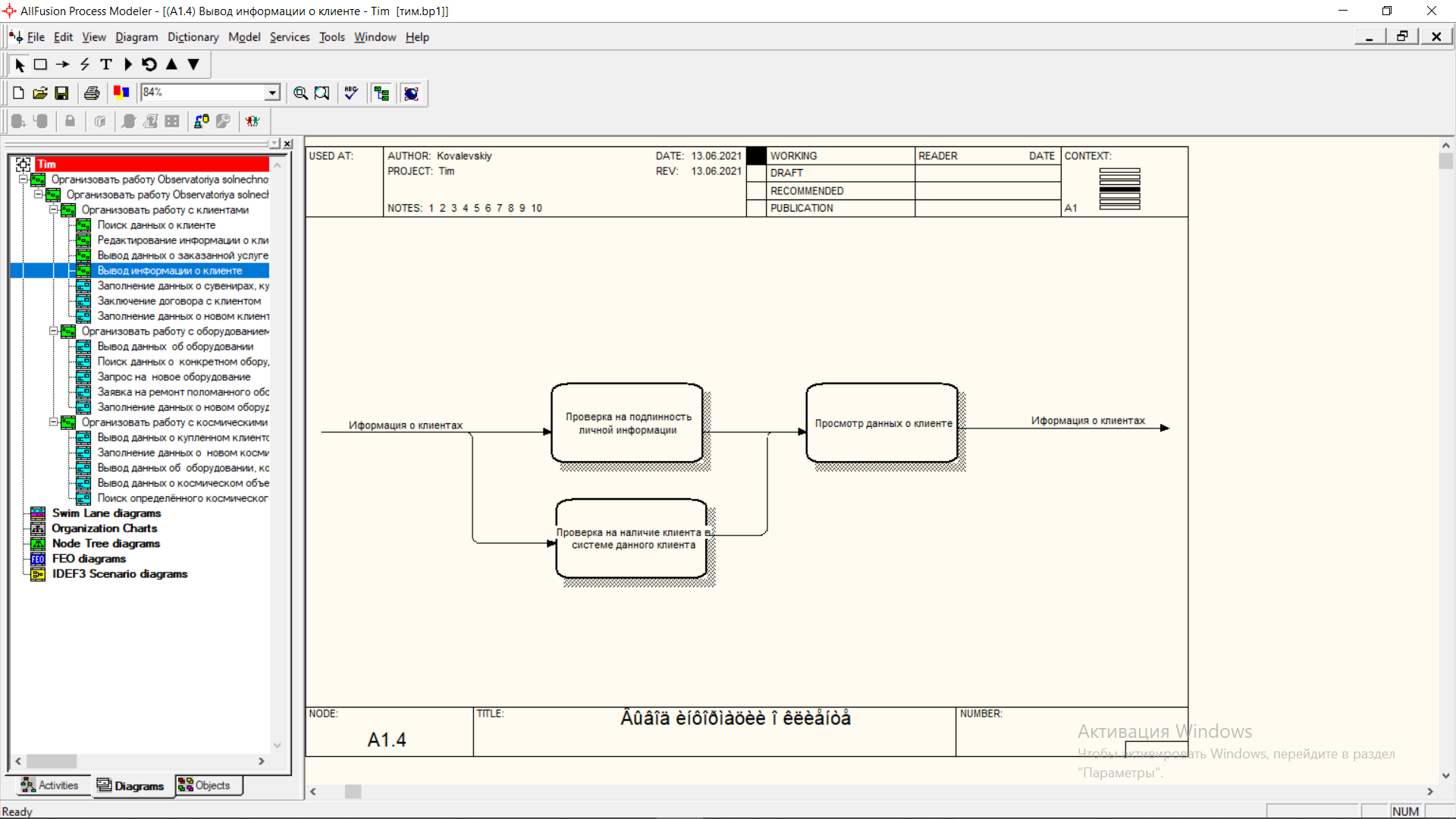


Рисунок 7 – декомпозиция процесса «Вывод информации о клиенте»

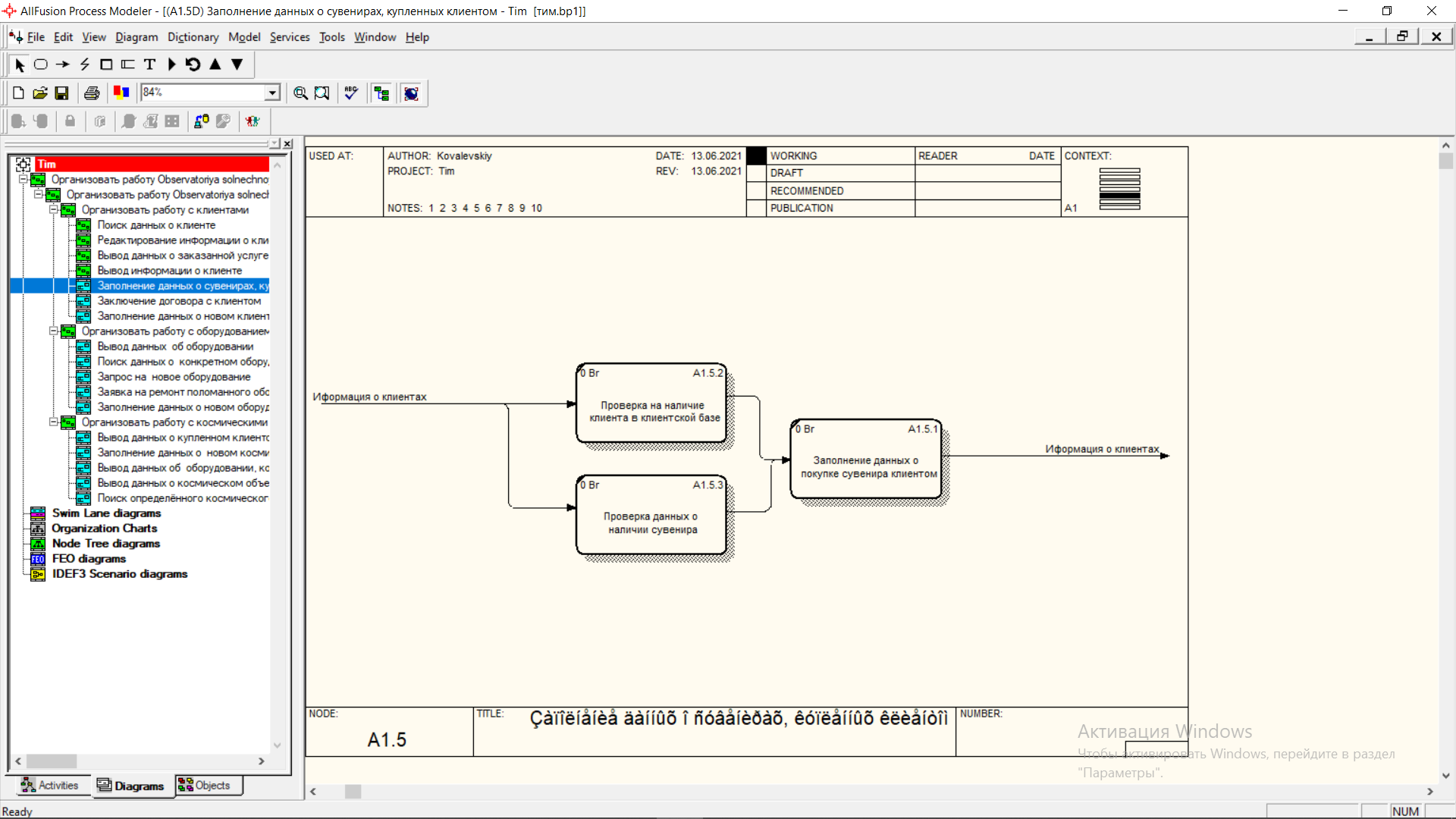


Рисунок 8 – декомпозиция процесса «Заполнение данных о сувенирах, купленных клиентом»

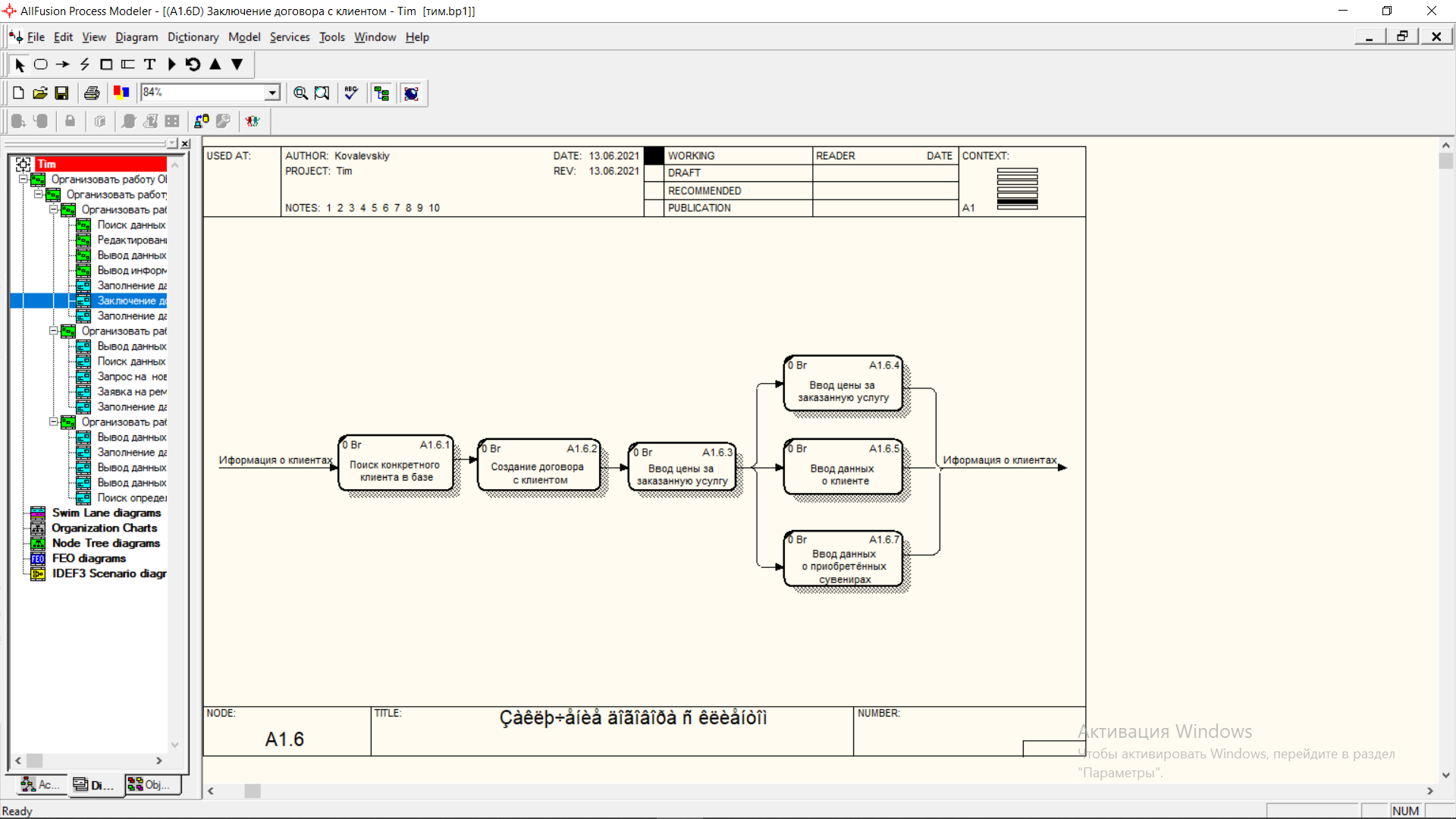


Рисунок 9 – декомпозиция процесса «Заключение договоров с клиентом»

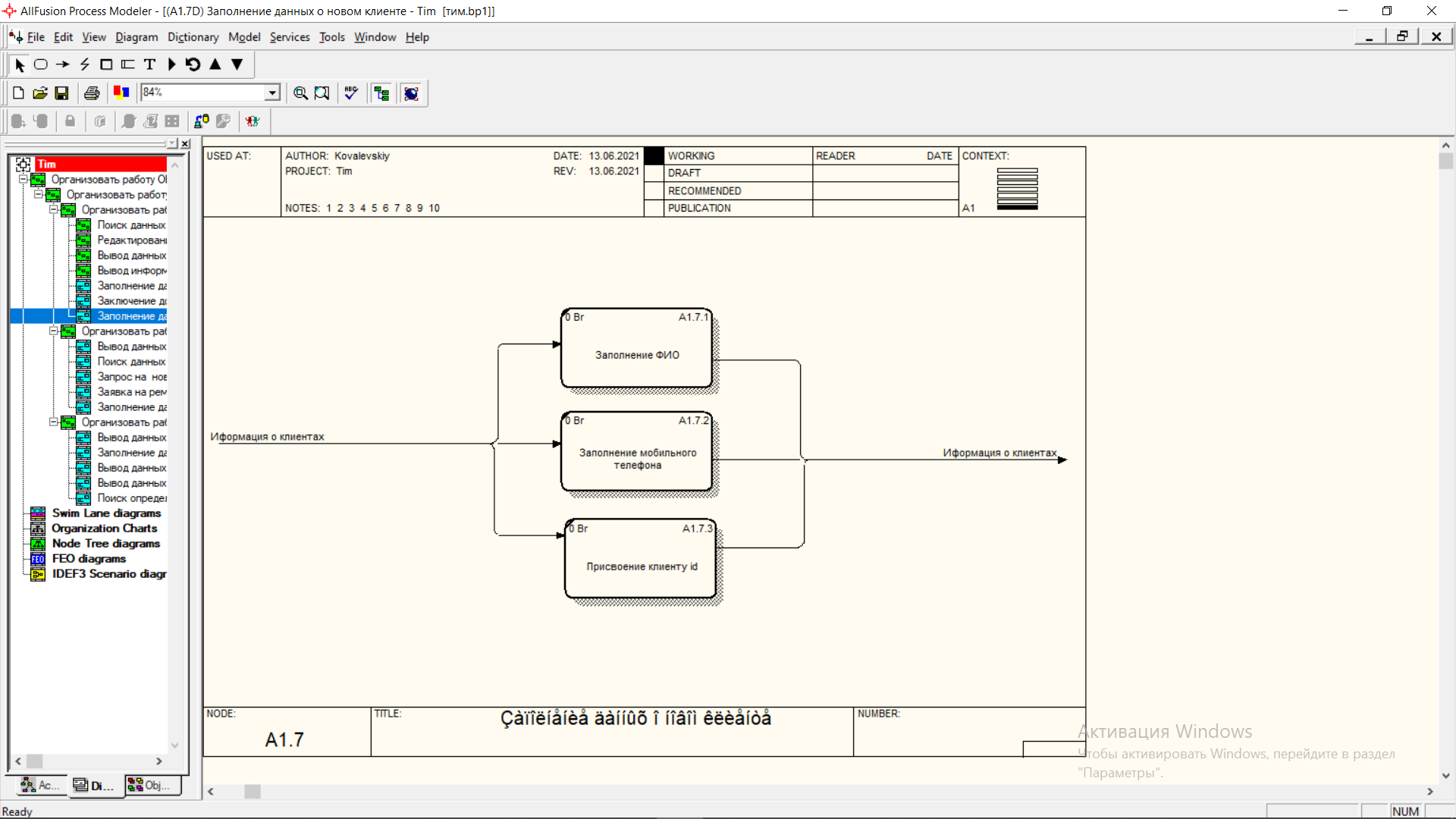


Рисунок 10 – декомпозиция процесса «Заполнение данных о новом клиенте»

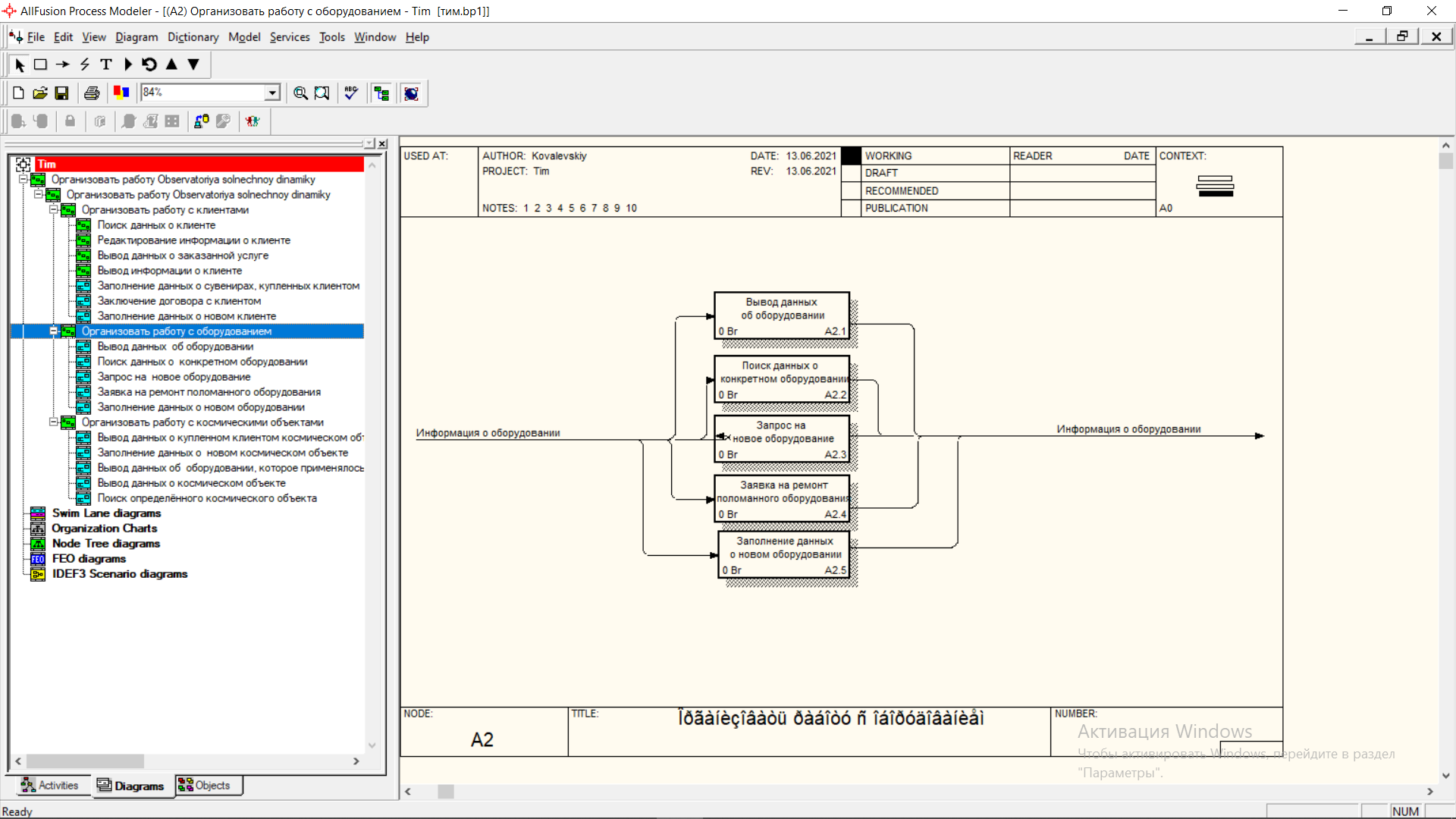


Рисунок 11 – декомпозиция процесса «Организовать работу с оборудованием»

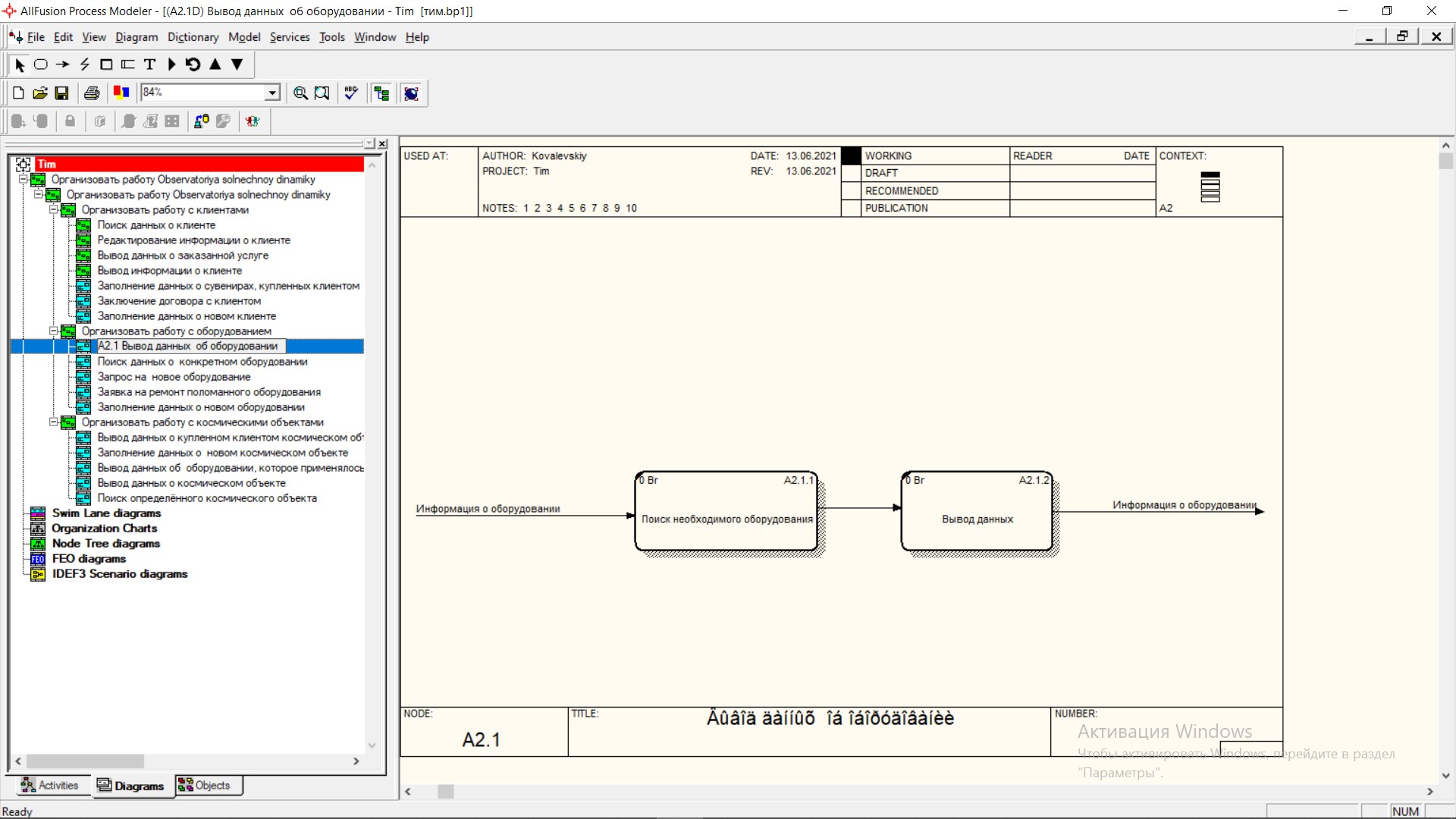


Рисунок 12 – декомпозиция процесса «Вывод данных об оборудовании»

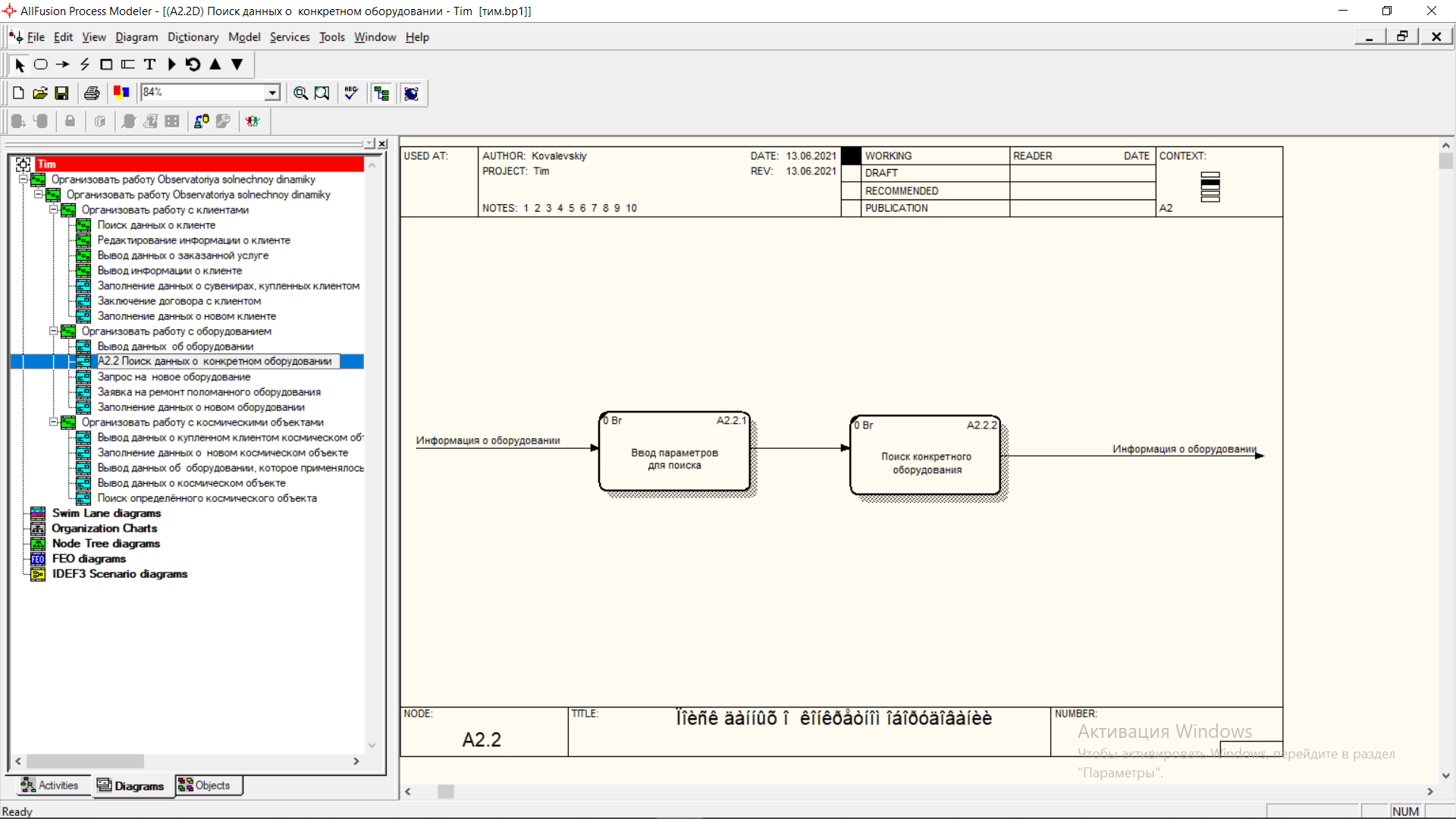


Рисунок 13 – декомпозиция процесса «Поиск данных о конкретном оборудовании»

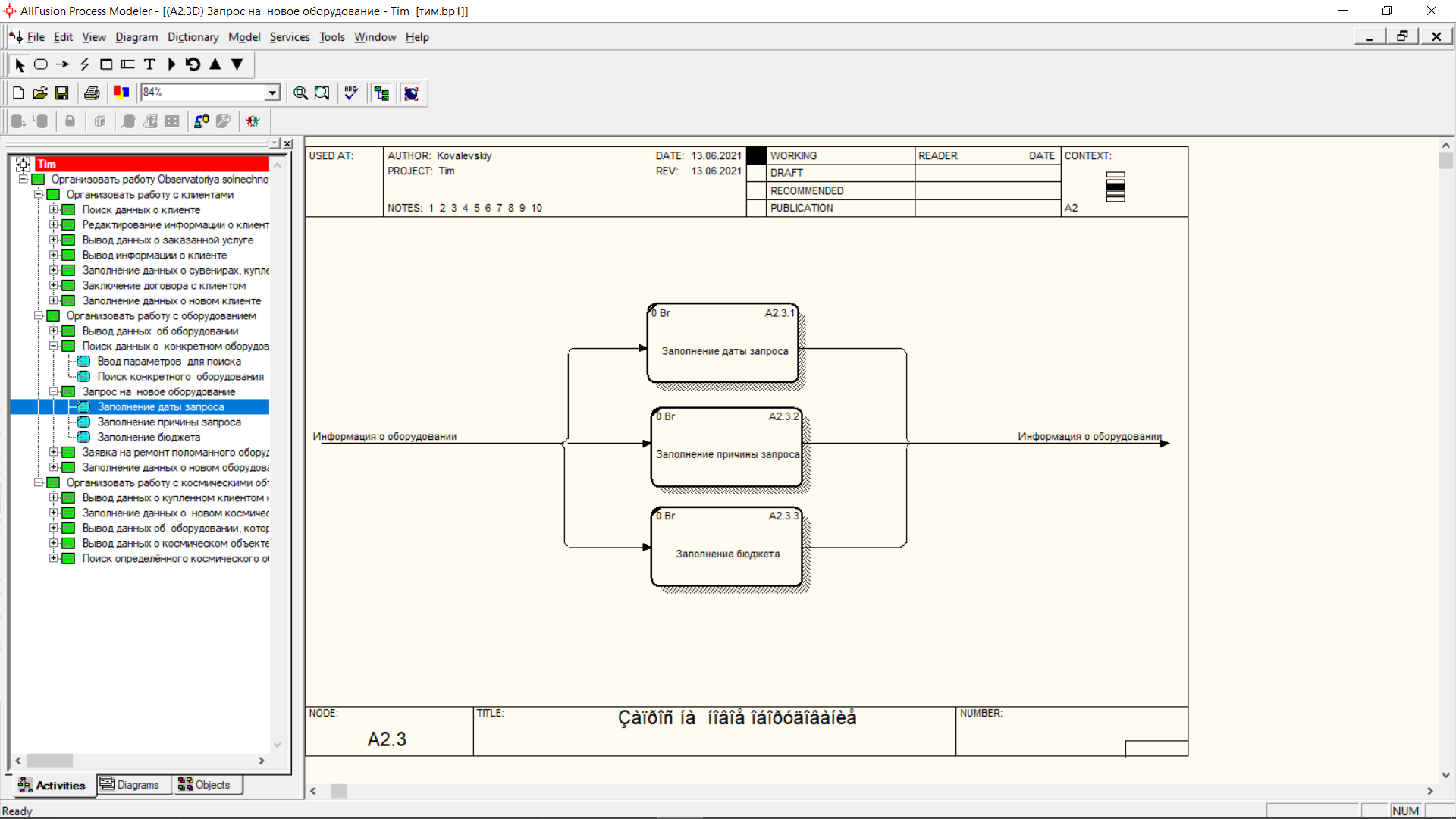


Рисунок 14 – декомпозиция процесса «Запрос на новое оборудование»

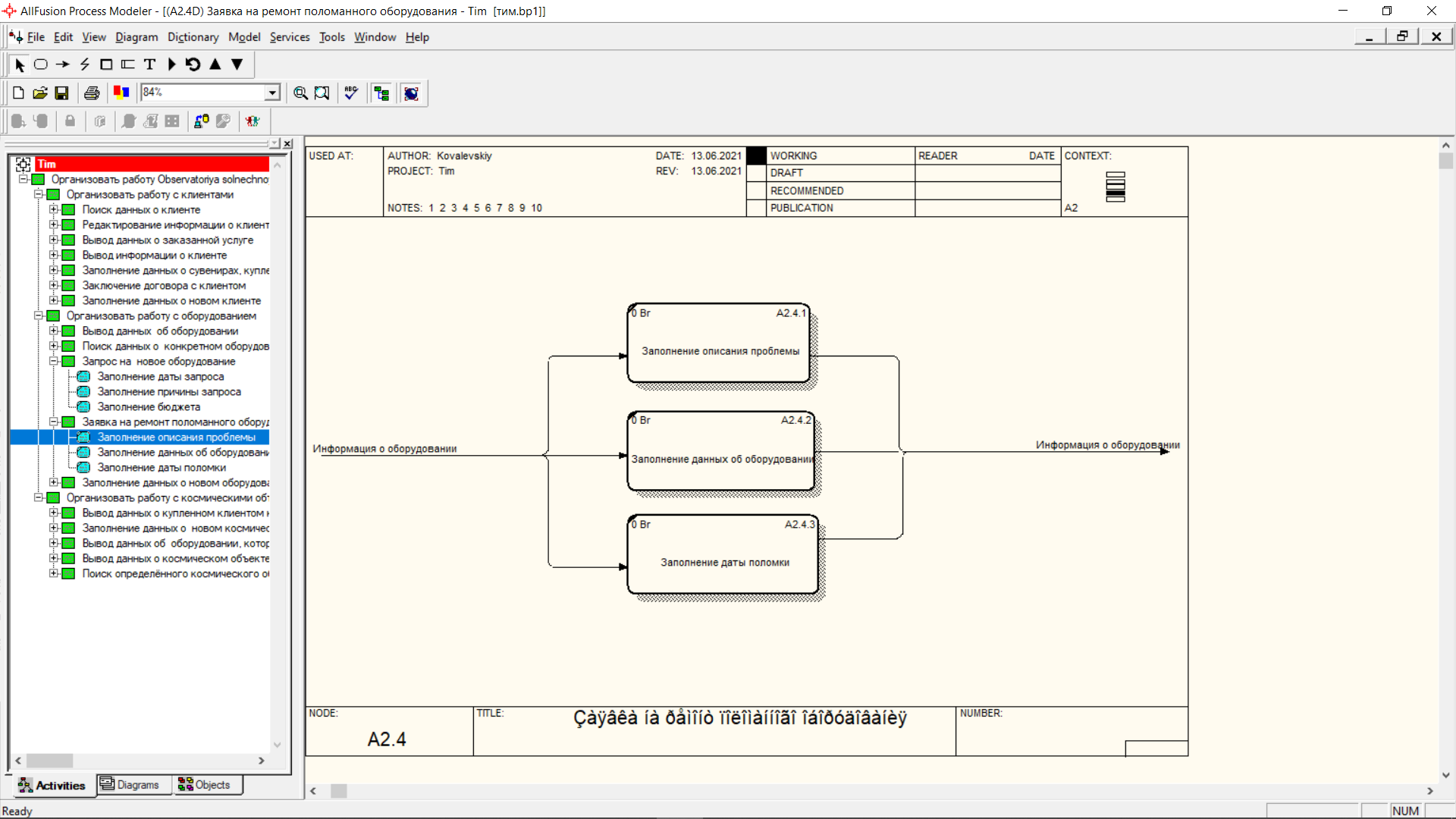


Рисунок 15 – декомпозиция процесса «Заявка на ремонт поломанного оборудования»

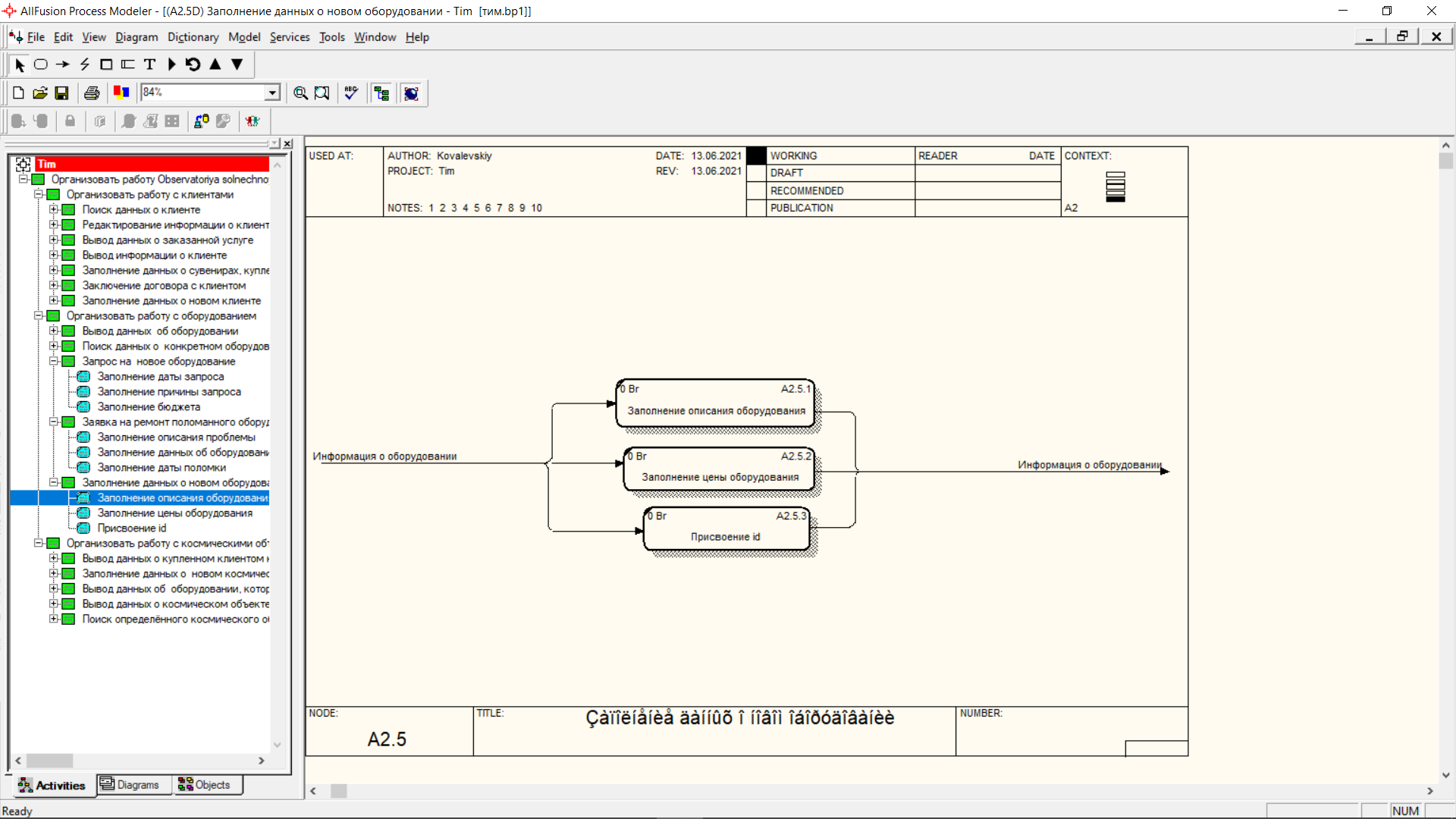


Рисунок 16 – декомпозиция процесса «Заполнение данных о новом оборудовании»

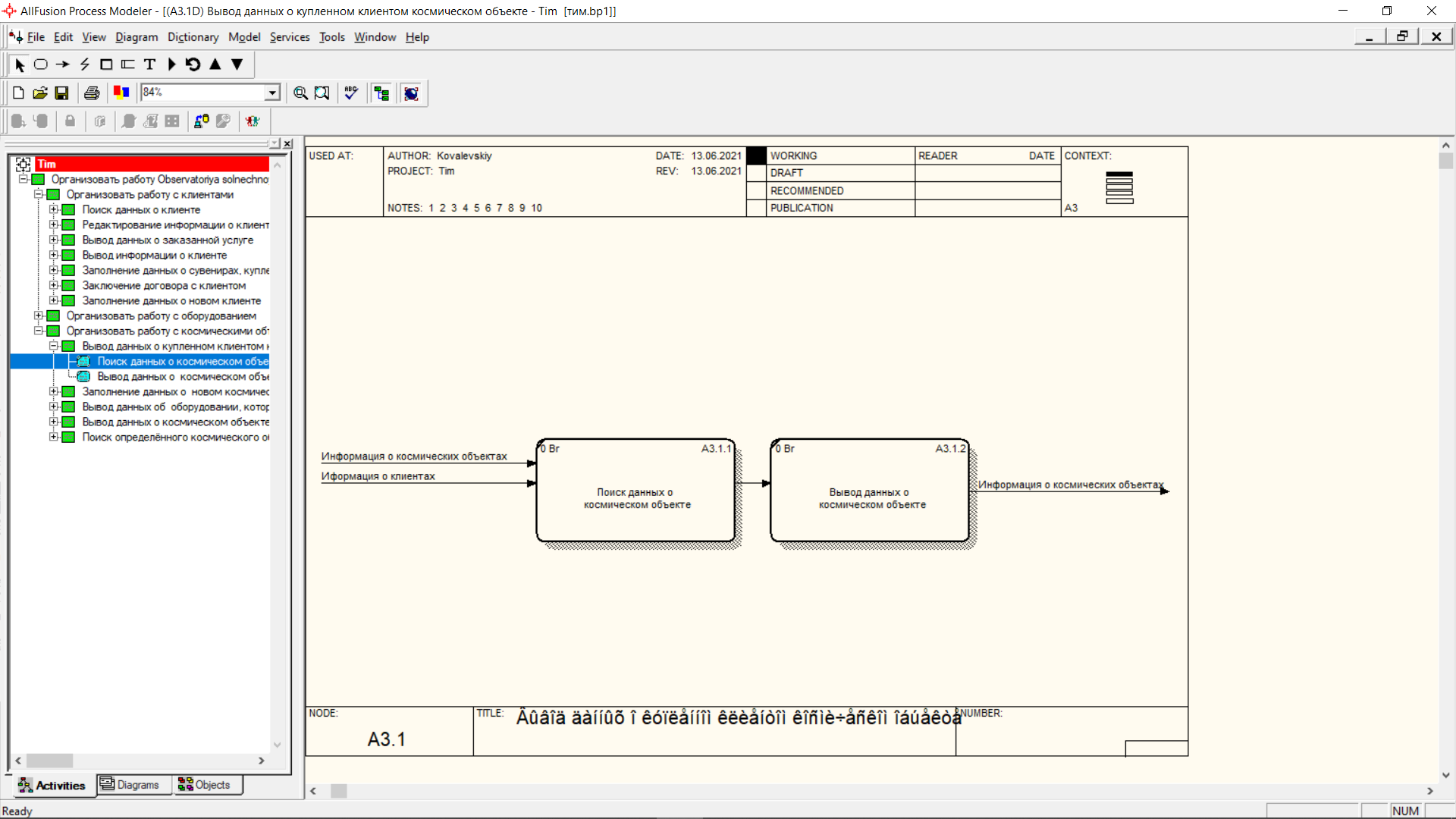


Рисунок 17 – декомпозиция процесса «Вывод данных о купленном клиентом космическом объекте»

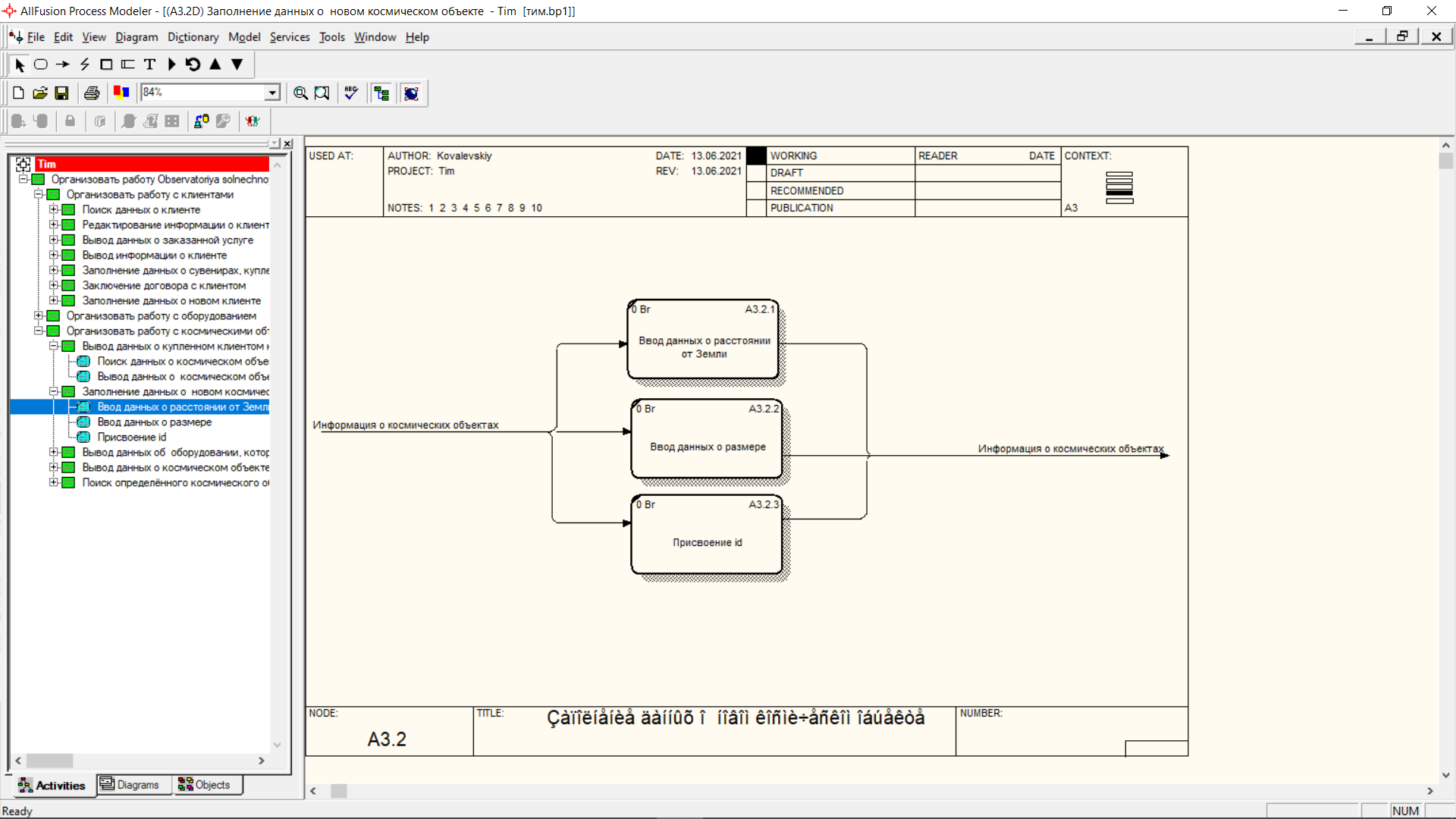
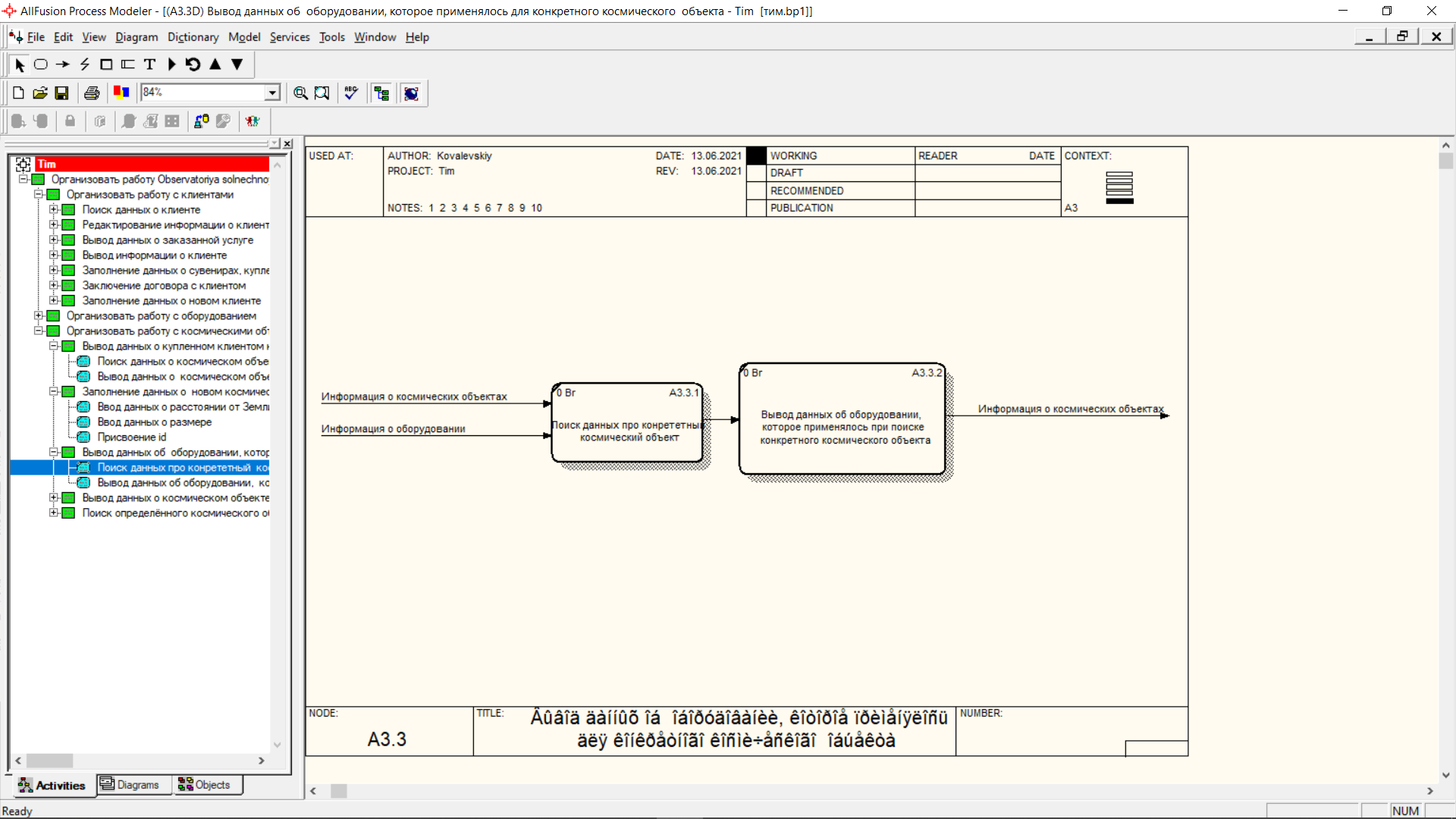


Рисунок 18 – декомпозиция процесса «Заполнение данных о новом космическом объекте»

 Рисунок 19 – декомпозиция процесса «Вывод данных об оборудовании, которое применялось для конкретного космического объекта»

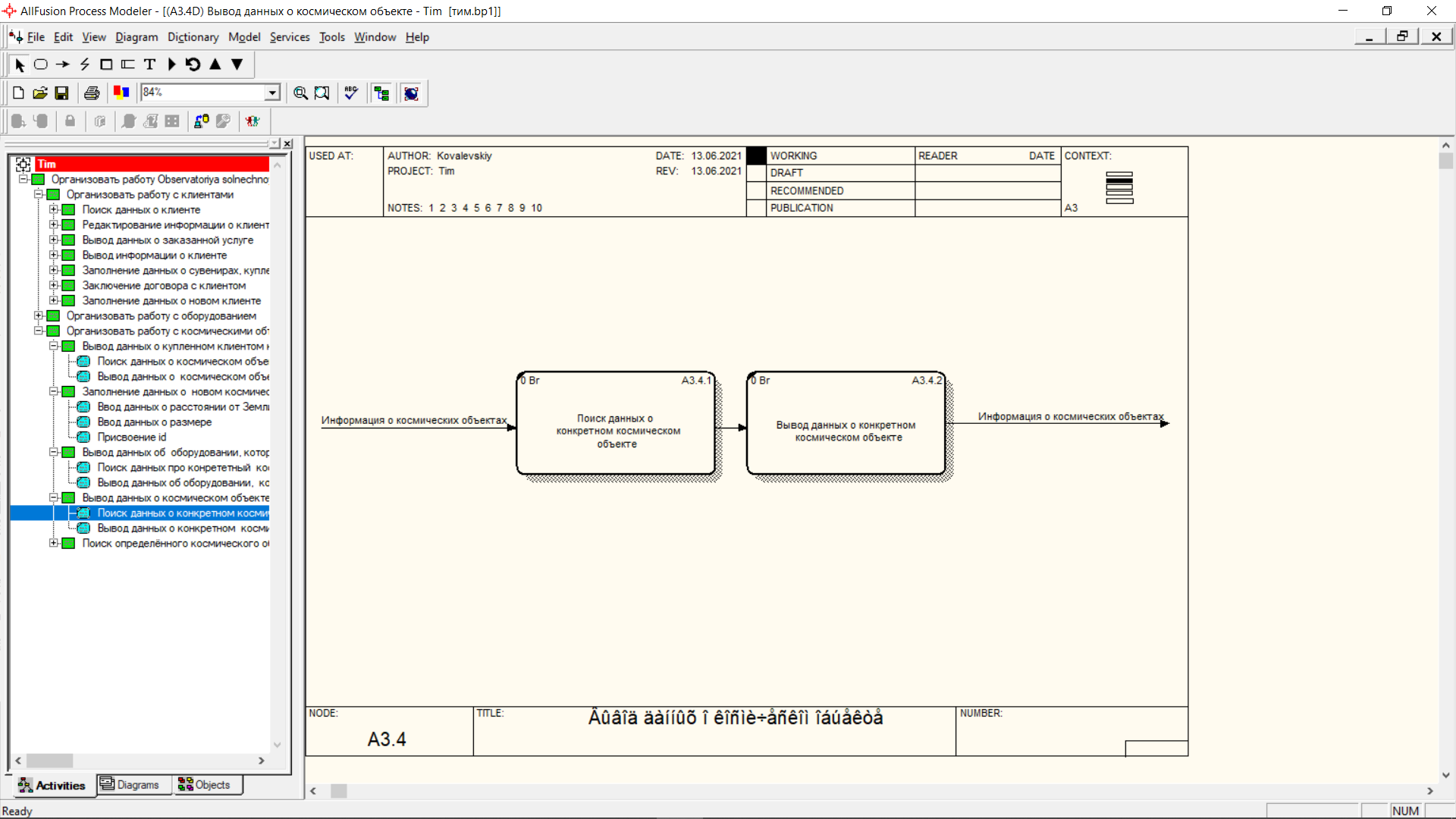


Рисунок 20 – декомпозиция процесса «Вывод данных о космическом объекте»

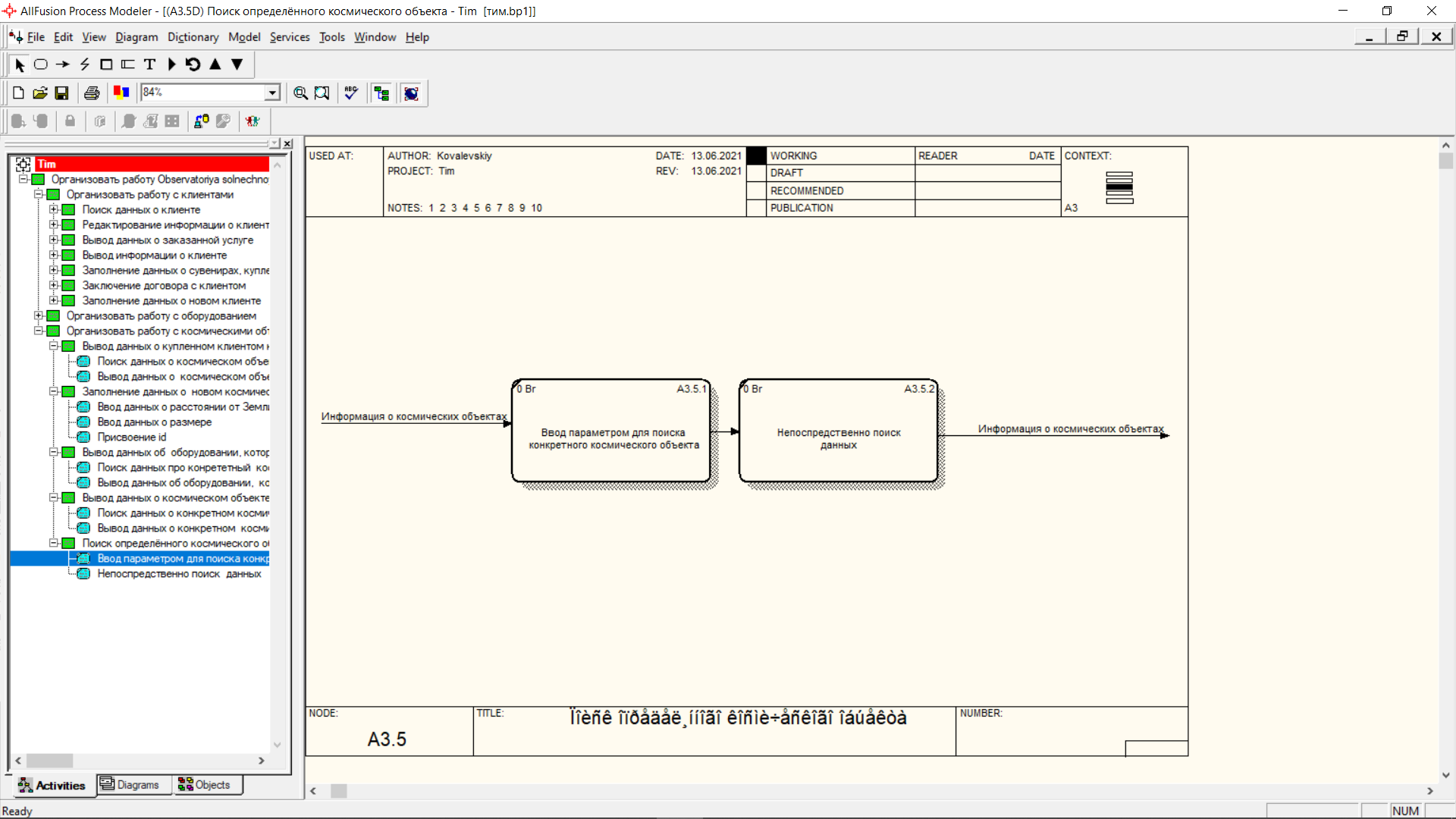


Рисунок 21 – декомпозиция процесса «Поиск определённого космического объекта»

# Диаграмма дерева узлов

На рисунке 22 и 23 представлена диаграмма дерева узлов.

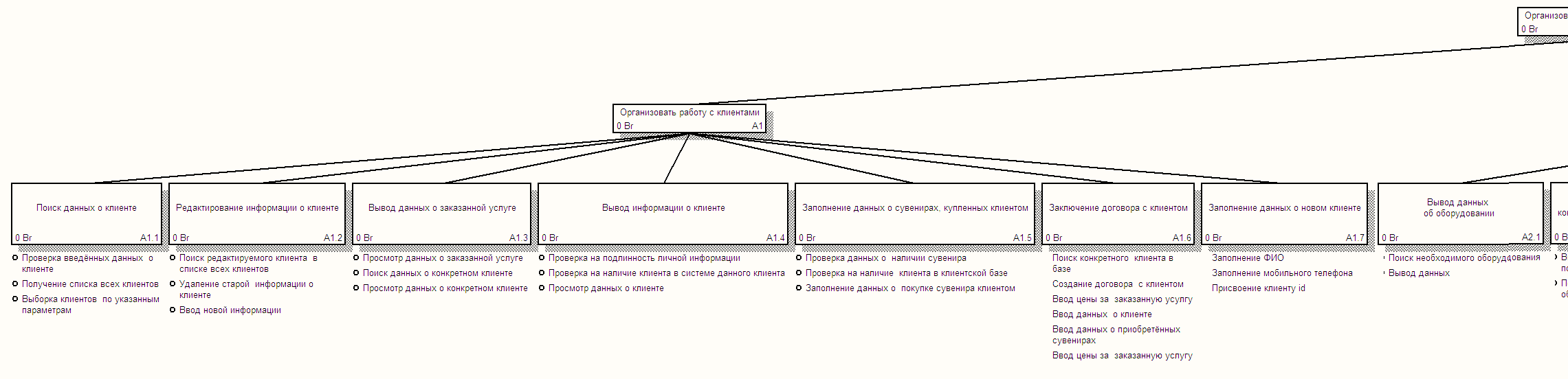


Рисунок 22 – Диаграмма дерева узлов

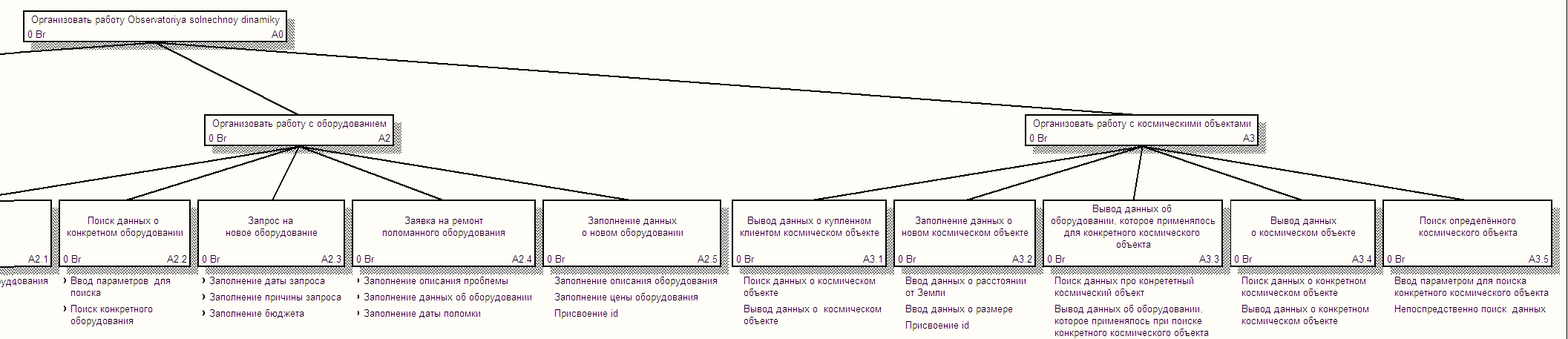


Рисунок 23 – Диаграмма дерева узлов

# Информационная модель

# Идентификация сущностей и связей. ER-диаграмма логического уровня

Для отображения информационной модели рассматриваемого процесса используются следующие сущности:

1. Договоры с клиентами – данная сущность хранит информацию о дате заключения договора.
2. Должность – данная сущность хранит информацию о всех задействованных должностях.
3. Клиенты – данная сущность хранит подробную информацию о каждом клиенте Обсерватории.
4. Космические объекты – данная сущность хранит полную информацию о космических объектах.
5. Оборудование – данная сущность хранит информацию о названиях и состоянии оборудования.
6. Отчеты – данная сущность хранит информацию о времени создания отчета.
7. Отчеты космических объектов – хранит информацию о ключах(id).
8. Состояние оборудования – данная сущность хранит информацию описания проблемы и причины списания оборудования.
9. Сотрудники – данная сущность хранит информацию о каждом сотруднике Обсерватории, такая как контактный номер, стаж, ФИО и информация закрепленная за ним – это отчеты, договора.
10. Списанное оборудование – данная сущность хранит информацию о причинах списания оборудования.
11. Сувениры – данная сущность хранит информацию о сувенирах магазина в Обсерватории.
12. Услуга – данная сущность хранит информация о различных услугах и ее стоимость.

Связь между сущностями:

* Связь между сущностями Услуга и Договоры с клиентами идентифицирующая, так как для оформления договора с клиентами необходима информация об услуге. Тип связи один ко многим, так как одна услуга может быть указана во многих договорах.
* Связь между сущностями Клиенты и Договоры с клиентами неиндентифицирующая. Тип связи один ко многим, так как договор с клиентами можно идентифицировать без клиента.
* Связь между сущностями Клиенты и Сувениры идентифицирующая, так ка для продажи сувениров необходима информация о клиенте. Тип связи один ко многим, так как один клиент может купить много сувениров.
* Связь между сущностями Сотрудник и Договор с клиентами неиндентифицирующая, так как один договор можно идентифицировать без сотрудника. Тип связи один ко многим.
* Связь между сущностями Должность и Сотрудники неиндентифицирующая, так как сотрудника можно идентифицировать без. Тип связи один к одному. Так как на одного сотрудника приходится одна должность.
* Связь между сущностями Сотрудники и Отчёты неиндентифицирующая, так как отчет можно идентифицировать без сотрудника. Тип связи один ко многим. Так как один сотрудник заведует и составляет отчёты.
* Связь между сущностями Отчёты и Космические объекты неиндентифицирующая, так как космические объекты можно идентифицировать без отчета. Тип связи многие ко многим, так как про один космический объект может быть составлено много отчётов, а в одном отчёте может быть много космических объектов.
* Связь между сущностями Состояние оборудования и Списанное оборудование неиндентифицирующая, так как списанное оборудование можно идентифицировать без состояния оборудования. Тип связи один к одному, так как исходя из одного состояния может быть списано одно оборудование.
* Связь между сущностями Сотрудник и Космические объекты неиндентифицирующая, так как космические объекты можно идентифицировать без сотрудника. Тип связи один ко многим, так как один сотрудник заведует космическими объектами.
* Связь между сущностями Сотрудники и Оборудование неиндентифицирующая, так как оборудование можно идентифицировать без сотрудника. Тип связи один ко многим, так как один сотрудник заведует всем оборудованием.
* Связь между сущностями Оборудование и Состояние оборудования неиндентифицирующая, так как состояние оборудования можно идентифицировать без оборудования. Тип связи один к одному, так как у одного оборудования может быть всего одно состояние.



Рисунок 24 – ER диаграмма логического уровня

# ER-диаграмма физического уровня. Ограничения ссылочной целостности. Переопределение триггеров. Индексирование отношений

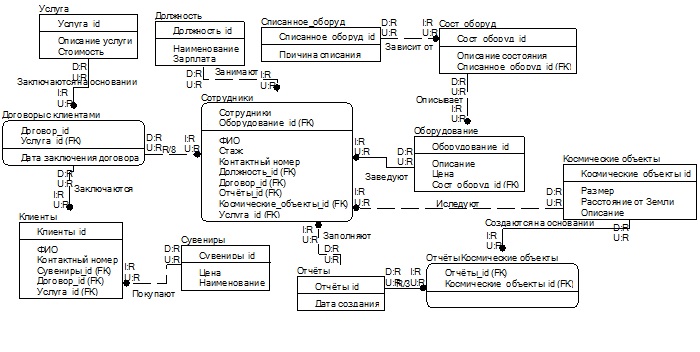


Рисунок 25 – ER диаграмма физического уровня

Реализация ссылочной целостности

При добавлении новой записи в зависимую таблицу значение внешнего ключа должно присутствовать в главной таблице.

При изменении информации о сотруднике из таблицы «Сотрудник» в таблицах «Должность», «Отчёты», «Договоры с клиентами», «Космические объекты» информация будет меняться автоматически.

При изменении данных в таблице «Должность» из таблицы «Отделы» в таблице «Сотрудник» информация будет автоматически меняться, удалять запрещено при наличии связанной записи.

При изменении информации о каком-либо клиенте из таблицы «Клиенты» в таблице «Договоры с клиентами» информация будет меняться автоматически.

При изменении информации в таблице «Оборудование» информация в таблицах «Состояние оборудования» будет меняться автоматически.

При изменении данных в таблице «Состояние оборудования» информация в таблице «Списанное оборудование» будет автоматически изменено.

При удалении информации о сотруднике в таблице «Сотрудник» информация о нём в таблице «Договоры с клиентами» сохранится.

В таблице «Клиент» при изменении данных происходит каскадное обновление. Разрешается удаление информации только в случае, когда на данную информацию нет ссылок в других связанных таблицах.

В таблице «Космические объекты» при изменении данных происходит каскадное обновление. Разрешается удаление информации только в случае, когда на данную информацию нет ссылок в других связанных таблицах.

В таблице «Списанное оборудование» запрещается удаление и изменение записей.

В таблице «Договоры с клиентом» запрещается удаление и изменение записей.

В таблице «Заказ» разрешается обновление.

В таблице «Сост. Оборудование» при изменении данных происходит каскадное обновление. Разрешается удаление информации только в случае, когда на данную информацию нет ссылок в других связанных таблицах.

# Определение представлений, хранимых процедур серверной компоненты, ER-диаграмма в режиме отображения представлений

В приложении были использованы представления, представленные в пункте 3.1.

Хранимые процедуры, использованные в проекте, описаны в пункте 3.3.

# Верификация спроектированной логической модели

Верификация спроектированной логической системы происходит посредствам взаимодействия программ Erwin и BPwin. Результат программ можно увидеть в таблице 1.

Результат связывания объектов модели процессов отображен в отчете 1.

| **Arrow Name** | **Entity Name** | **Attribute Name** |
| --- | --- | --- |
| Информация о космических объектах | Должность | Должность\_id |
|  |  | Зарплата |
|  |  | Наименование |
|  |  |  |
|  | Космические объекты | Космические\_объекты\_id |
|  |  | Описание |
|  |  | Размер |
|  |  | Расстояние от Земли |
|  | Отчёты | Дата создания |
|  |  | Отчёты\_id |
|  | Отчёты Космические объекты | Космические\_объекты\_id |
|  |  | Отчёты\_id |
|  | Сотрудники | Договор\_id |
|  |  | Должность\_id |
|  |  | Контактный номер |
|  |  | Космические\_объекты\_id |
|  |  | Оборудование\_id |
|  |  | Отчёты\_id |
|  |  | Стаж |
|  |  | ФИО |
| Информация о оборудовании | Оборудование | Оборудование\_id |
|  |  | Описание |
|  |  | Сост\_оборуд\_id |
|  |  | Цена |
|  | Сост\_оборуд | Описание состояния |
|  |  | Сост\_оборуд\_id |
|  |  | Списанное\_оборуд\_id |
|  | Списанное\_оборуд | Причина списания |
|  |  | Списанное\_оборуд\_id |
| Иформация о клиентах | Договоры с клиентами | Дата заключения договора |
|  |  | Договор\_id |
|  |  | Услуга\_id |
|  | Клиенты | Договор\_id |
|  |  | Клиенты\_id |
|  |  | Контактный номер |
|  |  | Сувениры\_id |
|  |  | ФИО |
|  | Сувениры | Наименование |
|  |  | Сувениры\_id |
|  |  | Цена |
|  |  |  |
|  |  |  |

# Реализация системы

# T-SQL-определения регламентированных запросов

Представление, позволяющее получить информацию о сувенирах и услугах.

CREATE VIEW AboutSourvenirService AS

SELECT dbo.Услуга.Стоимость,

dbo.Услуга.Описание\_услуги,

dbo.Сувениры.Наименование,

dbo.Сувениры.Цена

FROM

dbo.Услуга INNER JOIN dbo.Сувениры

on dbo.Услуга.Услуга\_id = dbo.Сувениры.Сувениры\_id

Представление, позволяющее получить информацию список должностей сотрудников.

CREATE VIEW ListPositionEmployee AS

SELECT dbo.Должность.Наименование

FROM

dbo.Должность INNER JOIN dbo.Сотрудники

on dbo.Сотрудники.Контактный\_номер != null

Представление, позволяющее получить информацию об отчетах космических объектах.

CREATE VIEW ReportingSpaceObjects AS

SELECT dbo.Космические\_объекты.[Размер км],

dbo.Отчёты.Дата\_создания

FROM

dbo.Космические\_объекты,

dbo.Отчёты,

(SELECT \*

FROM dbo.Клиенты, dbo.Отчёты

WHERE dbo.Отчёты.Дата\_создания > '2012-06-18 10:34:09.000'

) as reporting

WHERE dbo.Отчёты.Отчёты\_id > 6

Представление, позволяющее получить информацию о сортированных сувенирах по цене.

CREATE VIEW GetMaxPriceSourvenirs AS

SELECT dbo.Сувениры.Наименование,

dbo.Сувениры.Цена

FROM dbo.Сувениры ORDER BY dbo.Сувениры.Цена ASC

Представление, позволяющее получить информацию о списанном оборудовании и причины ее списания.

CREATE VIEW AboutEquipment AS

SELECT dbo.Сост\_оборуд.Описание\_состояния,

dbo.Списанное\_оборуд.Причина\_списания

FROM dbo.Списанное\_оборуд INNER INTO dbo.Сост\_оборуд

ON dbo.Списанное\_оборуд.Списанное\_оборуд\_id > 6

Представление, позволяющее получить информацию поиска должности и паттерну.

CREATE VIEW SerachByPattern AS

SELECT dbo.Должность.Наименование

FROM dbo.Должность WHERE dbo.Должность.Наименование LIKE '[И,К,С]%'

Представление, позволяющее получить информацию о сувенирах, у которых цена выше средней и отсортированные по убыванию по цене.

CREATE VIEW OrderSourvenirByPrice AS

SELECT dbo.Сувениры.Наименование,

dbo.Сувениры.Цена

FROM dbo.Сувениры

WHERE dbo.Сувениры.Цена >

((SELECT SUM(dbo.Сувениры.Цена)) / (SELECT AVG(dbo.Сувениры.Цена)))

ORDER BY dbo.Сувениры.Цена DESC

Представление, позволяющее получить информацию о клиентах, начиная с id = 1.

CREATE VIEW GetListClients AS

SELECT dbo.Клиенты.ФИО,

dbo.Клиенты.Контактный\_номер

FROM dbo.Клиенты

WHERE dbo.Клиенты.Клиенты\_id >= 1

Представление, позволяющее получить информацию расширенную об космических объектах, чьи размеры больше 10000000.

CREATE VIEW InfoSpace AS

SELECT dbo.Космические\_объекты.Описание,

dbo.Космические\_объекты.[Размер км],

dbo.Космические\_объекты.[Расстояние\_от\_Земли св.лет]

FROM dbo.Космические\_объекты FULL OUTER JOIN dbo.Отчёты

ON dbo.Отчёты.Отчёты\_id = dbo.Космические\_объекты.Космические\_объекты\_id

WHERE dbo.Космические\_объекты.[Размер км] > 100000000

Представление, позволяющее получить информацию о сувенирах, у которых присутствует цена.

CREATE VIEW AboutServis AS

SELECT \*

FROM dbo.Услуга

WHERE dbo.Услуга.Стоимость NOT LIKE NULL

Представление, позволяющее получить информацию о отчетах, в которых присутствует даты создания отчета.

CREATE VIEW InfoReportNotNull AS

SELECT \*

FROM dbo.Отчёты

WHERE dbo.Отчёты.Дата\_создания IS NOT NULL

Представление, позволяющее получить информацию об оборудование, где присутствует наличие описания оборудования.

CREATE VIEW InfoStatement AS

SELECT \*

FROM dbo.Оборудование

WHERE dbo.Оборудование.Описание IS NOT NULL

Представление, позволяющее получить информацию об отчетах, в которых отображается информация о космических объектах.

CREATE VIEW GetIDReporting AS

SELECT \*

FROM dbo.Отчёты\_Космические\_объекты

WHERE dbo.Отчёты\_Космические\_объекты.Космические\_объекты\_id = dbo.Отчёты\_Космические\_объекты.Отчёты\_id

Представление, позволяющее получить информацию полную о клиентах, которые указали свой номер телефона.

CREATE VIEW GetAllManadgers AS

SELECT \*

FROM dbo.Должность FULL OUTER JOIN dbo.Клиенты

ON dbo.Клиенты.Контактный\_номер NOT LIKE NULL

Представление, позволяющее получить информацию о имеющийся должностях которые начинается с паттерна «Инжинер».

CREATE VIEW SearchPatternByEmployee AS

SELECT \*

FROM dbo.Должность

WHERE dbo.Должность.Наименование LIKE 'Инженер'

# T-SQL-определения триггеров

Триггер на вставку записей отчетов с проверкой

CREATE TRIGGER [Отчёта - Вставка записей (с проверкой)]

ON Отчёт

AFTER INSERT

AS

IF (SELECT Дата\_создания FROM Отчёт) >= getDate()

BEGIN

INSERT INTO HistoryActionOperation (тип\_операции, таблица, идентификатор\_записи, дата)

VALUES ('Добавление', 'Отчёт', (SELECT Отчёты\_id FROM Отчёты), getDate())

END

ELSE

BEGIN

PRINT ('Ошибка. Некорректная ввод данных)

ROLLBACK TRANSACTION

END

Триггер на проверку корректности введенной цены больше 1000000

CREATE TRIGGER [Услуга - проверка корректной цены]

ON Услуга

AFTER INSERT

AS

IF EXISTS (SELECT \* FROM Услуга WHERE Стоимость > 1000000)

BEGIN

PRINT 'Ошибка. Некорректная ценна ввода'

ROLLBACK TRANSACTION

END

Триггер на проверку корректности введенной стоимости

CREATE TRIGGER [Услуга - проверка корректной цены меньшей 1]

ON Услуга

AFTER INSERT

AS

IF EXISTS (SELECT \* FROM Услуга WHERE Стоимость < 1)

BEGIN

PRINT 'Ошибка. Отрицательная цена ввода'

ROLLBACK TRANSACTION

END

Триггер на удаление услуги

CREATE TRIGGER [Услуга - Удаление записей]

ON Услуга

AFTER DELETE

AS

INSERT INTO HistoryActionOperation (тип\_операции, таблица, идентификатор\_записи, дата)

VALUES ('Удаление', 'Услуга', (SELECT Услуга\_id FROM Услуга), getDate())

Триггер на удаление сувенира

CREATE TRIGGER [Сувениры - Удаление записей]

ON Сувениры

AFTER DELETE

AS

INSERT INTO HistoryActionOperation (тип\_операции, таблица, идентификатор\_записи, дата)

VALUES ('Удаление', ' Сувениры ', (SELECT Сувениры \_id FROM Сувениры), getDate())

Триггер на удаление оборудования

CREATE TRIGGER [Оборудование - Удаление записей]

ON Оборудование

AFTER DELETE

AS

INSERT INTO HistoryActionOperation (тип\_операции, таблица, идентификатор\_записи, дата)

VALUES ('Удаление', ' Оборудование ', (SELECT Оборудование \_id FROM Оборудование), getDate())

Триггер на встаку оборудования

CREATE TRIGGER [Оборудование - Измененеие записей]

ON Оборудование

AFTER UPDATE

AS

INSERT INTO HistoryActionOperation (тип\_операции, таблица, идентификатор\_записи, дата)

VALUES ('Изменение', ' Оборудование ', (SELECT Оборудование \_id FROM Оборудование),getDate())

Триггер на вставку клиентов

CREATE TRIGGER [Клиенты - Измененеие записей]

ON Клиенты

AFTER UPDATE

AS

INSERT INTO HistoryActionOperation (тип\_операции, таблица, идентификатор\_записи, дата)

VALUES ('Изменение', 'Клиенты', (SELECT Клиенты\_id FROM Клиенты), getDate())

# T-SQL-определения хранимых процедур

Хранимая процедура для показа краткой информации об оборудовании.

CREATE PROCEDURE EquipmentSmmary

AS

SELECT Оборудование.Описание as descrition, Оборудование.Цена, Оборудование.Оборудование\_id

FROM Оборудование

Хранимая процедура для получения списка сувениров в заданном интервале цены.

CREATE PROCEDURE get\_souvenirs @get\_price VARCHAR(50)

AS

SELECT \*

FROM Сувениры ORDER BY

CASE @get\_price

WHEN 'маленькая сумма'

THEN

(SELECT SUM(Сувениры.Цена) as price

FROM Сувениры

GROUP BY Сувениры.Цена

HAVING SUM(Сувениры.Цена) < 14)

WHEN 'средняя сумма'

THEN

(SELECT SUM(Сувениры.Цена) as price

FROM Сувениры

GROUP BY Сувениры.Цена

HAVING SUM(Сувениры.Цена) > 14 and SUM(Сувениры.Цена) < 100)

WHEN 'Большая сумма'

THEN

(SELECT SUM(Сувениры.Цена) as price

FROM Сувениры

GROUP BY Сувениры.Цена

HAVING SUM(Сувениры.Цена) > 100)

END

Хранимая процедура для получения контрактов, которые были заключены между определенными датами.

CREATE PROCEDURE ContractClientByDate @d1 DATETIME, @d2 DATETIME

AS

SELECT \*

FROM Договоры\_с\_клиентами

WHERE Договоры\_с\_клиентами.Дата\_заключения\_договора BETWEEN @d1 and @d2

Хранимая процедура для получения договора с клиентами, у которых есть услуга и сгруппировать все по услуге id.

CREATE PROCEDURE ContractClientByServiseID @serviseID smallint

AS

SELECT \*

FROM Договоры\_с\_клиентами

INNER JOIN Услуга

ON Услуга.Услуга\_id = @serviseID

GROUP BY Договоры\_с\_клиентами.Услуга\_id

Хранимая процедура для получения списка зарплат, которые превышают заданный параметр.

CREATE PROCEDURE PositionSalary @salary smallint

AS

SELECT Должность.Должность\_id, Должность.Зарплата, Должность.Наименование

FROM Должность

WHERE Должность.Зарплата > @salary

Хранимая процедура для поиска должности по названию.

CREATE PROCEDURE FindPositionSalary @title VARCHAR(50)

AS

SELECT Должность.Должность\_id, Должность.Зарплата, Должность.Наименование

FROM Должность

WHERE @title = Должность.Наименование

Хранимая процедура для поиска планет, которые находятся на определенном расстоянии.

CREATE PROCEDURE GetPlanetByDistance @distance int as

CASE @distance

WHEN 10

THEN

SELECT Космические\_объекты.[Расстояние\_от\_Земли св.лет] as length\_,

Космические\_объекты.Описание as title

FROM Космические\_объекты

GROUP BY Космические\_объекты.[Расстояние\_от\_Земли св.лет]

HAVING Космические\_объекты.[Расстояние\_от\_Земли св.лет] < @distance

WHEN 50

THEN

SELECT Космические\_объекты.[Расстояние\_от\_Земли св.лет] as length\_,

Космические\_объекты.Описание as title

FROM Космические\_объекты

GROUP BY Космические\_объекты.[Расстояние\_от\_Земли св.лет]

HAVING Космические\_объекты.[Расстояние\_от\_Земли св.лет] < @distance

WHEN 100

THEN

SELECT Космические\_объекты.[Расстояние\_от\_Земли св.лет] as length\_,

Космические\_объекты.Описание as title

FROM Космические\_объекты

GROUP BY Космические\_объекты.[Расстояние\_от\_Земли св.лет]

HAVING Космические\_объекты.[Расстояние\_от\_Земли св.лет] < @distance

ELSE 'No planets'

END CASE;

Хранимая процедура для того, что бы выбрать космические объекты по их размерам.

CREATE PROCEDURE GetMinOrMaxPlanetSize @size int

AS

SELECT Космические\_объекты.[Размер км], Космические\_объекты.Описание,

CASE

WHEN Космические\_объекты.[Размер км] = min(Космические\_объекты.[Размер км])

THEN 'Найдена минимальный размер планеты'

WHEN Космические\_объекты.[Размер км] = max(Космические\_объекты.[Размер км])

THEN 'Найдена максимальный размер планеты'

END AS texting

FROM Космические\_объекты;

Хранимая процедура для выборки всего оборудования и его состояния, сгруппировать по оборудованию id.

CREATE PROCEDURE EquimentState @stateID smallint

AS

SELECT Оборудование.Описание, Оборудование.Сост\_оборуд\_id, COUNT(Оборудование.Сост\_оборуд\_id)

FROM Оборудование INNER JOIN Сост\_оборуд ON Оборудование.Сост\_оборуд\_id = Сост\_оборуд.Сост\_оборуд\_id

GROUP BY Оборудование.Сост\_оборуд\_id

HAVING COUNT(Оборудование.Сост\_оборуд\_id)>1

Хранимая процедура для получения списка описания об оборудовании и отсортировать по алфавиту.

CREATE PROCEDURE SortDescriptionEquimentByAlph @type\_sort VARCHAR(15)

AS

SELECT Оборудование.Описание

CASE

WHEN @type\_sort = 'asc'

THEN SELECT Оборудование.Описание FROM Оборудование ORDER BY Оборудование.Описание ASC

WHEN @type\_sort = 'desc'

THEN SELECT Оборудование.Описание FROM Оборудование ORDER BY Оборудование.Описание DESC

END AS texting

FROM Оборудование

Хранимая процедура для получения списка отчетов, которые были созданы между определенных дат.

CREATE PROCEDURE get\_data\_reporting (d1 DATETIME, d2 DATETIME)

AS

SELECT \*

FROM Отчёты

WHERE Отчёты.Дата\_создания BETWEEN d1 and d2

GROUP BY Отчёты.Дата\_создания

Хранимая процедура для создания нового отчета, с предварительной проверкой на созданный уже раннее.

CREATE PROCEDURE CreateReport @id smallint, @dt\_tm datetime

AS

IF EXISTS (SELECT \* FROM Отчёты

WHERE Отчёты.Дата\_создания = @dt\_tm OR Отчёты.Отчёты\_id = @id

)

RETURN 0

INSERT dbo.Отчёты (Отчёты\_id, Дата\_создания)

VALUES(@id, @dt\_tm)

Хранимая процедура для обновления информации об оборудовании с предварительной проверкой на существование данной записи.

CREATE PROCEDURE UpdateStateEquiment @id smallint, @decsrp varchar(200), @new\_descrp VARCHAR(200)

AS

if EXISTS (SELECT \* FROM Сост\_оборуд

WHERE Сост\_оборуд.Сост\_оборуд\_id != @id

)

RETURN 0

UPDATE dbo.Сост\_оборуд

SET Описание\_состояния = @new\_decsrp

WHERE Описание\_состояния = @decsrp

Хранимая процедура для удаления записи состояния оборудования с предварительной проверкой на существование данной записи.

CREATE PROCEDURE DeleteStateEquiment @id smallint

AS

if EXISTS (SELECT \* FROM Сост\_оборуд

WHERE Сост\_оборуд.Сост\_оборуд\_id != @id

)

RETURN 0

DELETE FROM Сост\_оборуд

WHERE Сост\_оборуд.Сост\_оборуд\_id = @id

Хранимая процедура для поиска сотрудника по его стажу и вывод расширенной информации о нем с предварительной проверкой на существование данного сотрудника.

CREATE PROCEDURE EmployeeExperience @years int

AS

IF EXISTS (SELECT Сотрудники.id, Сотрудники.ФИО, Сотрудники.Стаж

FROM Сотрудники

WHERE Сотрудники.Стаж != @years

)

RETURN 0

SELECT Сотрудники.Договор\_id, Сотрудники.ФИО,

Сотрудники.Контактный\_номер, Сотрудники.Должность\_id,

Сотрудники.Оборудование\_id

FROM Сотрудники

INNER JOIN Должность ON Должность.Должность\_id = Сотрудники.Должность\_id

INNER JOIN Оборудование ON Оборудование.Оборудование\_id = Сотрудники.Оборудование\_id

GROUP BY Сотрудники.Договор\_id

Хранимая процедура для обновления записи о сотрудниках с предварительной проверкой на наличие данной записи.

CREATE PROCEDURE UpdateEmployeePhone @id smallint, @newphone VARCHAR(50)

AS

IF EXISTS(SELECT \* FROM Сотрудники WHERE Сотрудники.id = @id)

RETURN 0

UPDATE Сотрудники

SET Контактный\_номер = @newphone

WHERE id = @id

Хранимая процедура для выбора только уникальных записей списанного оборудоавния.

CREATE PROCEDURE GetUniqueBreakEquiment

AS

SELECT DISTINCT Списанное\_оборуд.Причина\_списания, Списанное\_оборуд.Списанное\_оборуд\_id

FROM Списанное\_оборуд

Хранимая процедура для добавления новой записи о списанном оборудовании с предварительной проверкой на наличие данной записи.

CREATE PROCEDURE InsertBreakEquiment @id smallint, @reason VARCHAR(50)

AS

IF EXISTS (SELECT \* FROM Списанное\_оборуд WHERE Списанное\_оборуд.Списанное\_оборуд\_id = @id)

RETURN 0

INSERT INTO Списанное\_оборуд(Списанное\_оборуд\_id, Причина\_списания)

VALUES (@id, @reason)

Хранимая процедура для обновления списанного оборудования с предварительной проверкой на наличие данной записи.

CREATE PROCEDURE UpdateBreakEquiment

@id smallint, @reason VARCHAR(100)

AS

IF not EXISTS (SELECT \* FROM Списанное\_оборуд WHERE Списанное\_оборуд.Списанное\_оборуд\_id = @id)

RETURN 0

UPDATE Списанное\_оборуд

SET Причина\_списания = @reason

WHERE Списанное\_оборуд\_id = @id

Хранимая процедура для обновления записи о сувенирах с предварительной проверкой на наличие записи.

CREATE PROCEDURE UpdateSouvenir

@id smallint, @price VARCHAR(100)

AS

IF not EXISTS (SELECT \* FROM Сувениры WHERE Сувениры\_id=@id)

RETURN 0

UPDATE Сувениры

SET Цена = @price

WHERE Сувениры\_id = @id

# T-SQL-определения курсоров

Курсор для просмотра должностей

DECLARE countJobs CURSOR

GLOBAL

KEYSET

TYPE\_WARNING

FOR

SELECT Должность\_id, Наименование, Зарплата FROM Должность

FOR READ ONLY

OPEN GLOBAL countJobs

DECLARE @@Counter int

SET @@Counter =@@CURSOR\_ROWS

Select @@Counter

CLOSE countJobs

DEALLOCATE countJobs

Курсор для просмотра договоров с клинтами

DECLARE countreport CURSOR

GLOBAL

KEYSET

TYPE\_WARNING

FOR

SELECT Договор\_id, Дата\_заключения\_договора, Услуга\_id FROM Договоры\_с\_клиентами

FOR READ ONLY

OPEN GLOBAL countreport

DECLARE @@Counter int

SET @@Counter =@@CURSOR\_ROWS

Select @@Counter

CLOSE countreport

DEALLOCATE countreport

Курсор для просмотра количества клиентов.

DECLARE countCLient CURSOR

GLOBAL

KEYSET

TYPE\_WARNING

FOR

SELECT Клиент\_id,ФИО,Контактный\_номер,Сувениры\_id,Договор\_id,Услуга\_id FROM Клиенты

FOR READ ONLY

OPEN GLOBAL countCLient

DECLARE @@Counter int

SET @@Counter =@@CURSOR\_ROWS

Select @@Counter

CLOSE countCLient

Курсор для подсчета количества оборудования

DEALLOCATE countCLient

DECLARE countStateThings CURSOR

GLOBAL

KEYSET

TYPE\_WARNING

FOR

SELECT Сост\_оборуд\_id,Описание\_состояния, Списание\_оборуд\_id FROM Сост\_оборудования

FOR READ ONLY

OPEN GLOBAL countStateThings

DECLARE @@Counter int

SET @@Counter =@@CURSOR\_ROWS

Select @@Counter

CLOSE countStateThings

DEALLOCATE countStateThings

Курсор для подсчета сувеноров

DECLARE countGifts CURSOR

GLOBAL

KEYSET

TYPE\_WARNING

FOR

SELECT Сувениры\_id, Цена,Наименование FROM Сувениры

FOR READ ONLY

OPEN GLOBAL countWork

DECLARE @@Counter int

SET @@Counter =@@CURSOR\_ROWS

Select @@Counter

CLOSE countGifts

DEALLOCATE countGifts

# Описание клиентских приложений

При запуске открывается окно авторизации, представленное на рисунке 25.

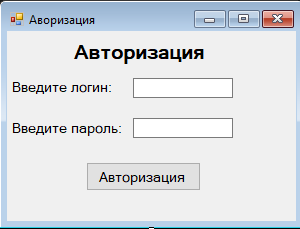


Рисунок 25 – Окно для авторизации пользователя

В данной системе пользователь может быть авторизован как клиент или как продавец. Продавцам, как и пользователям не доступен весь функционал системы. Основная форма приложения представлена на рисунке 31.

Формы добавления, редактирования и удаления данных в базу представлены на рисунках ниже.

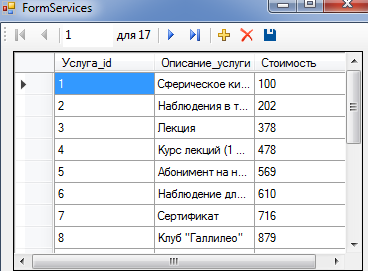


Рисунок 26 – Окно формы «Услуга»

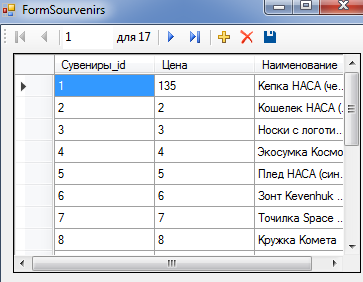


Рисунок 27 – Окно формы «Сувениры»

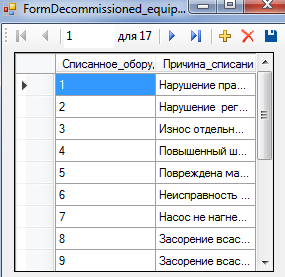


Рисунок 28 – Окно формы «»

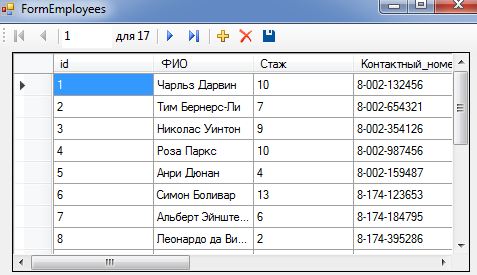


Рисунок 29 – Окно формы «Сотрудники»

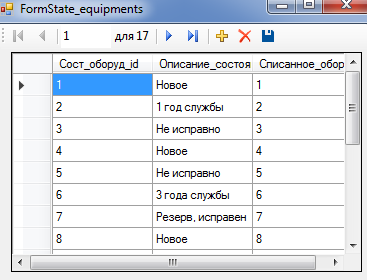


Рисунок 30 – Окно формы «Сотояние оборудования»

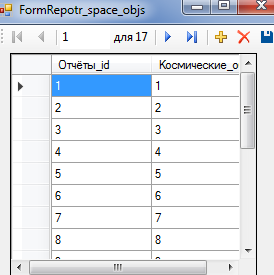


Рисунок 31 – Окно формы «Отчеты»

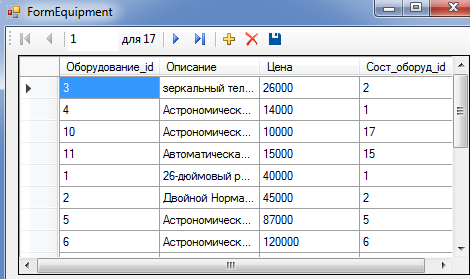


Рисунок 32 – Окно формы «Оборудование»

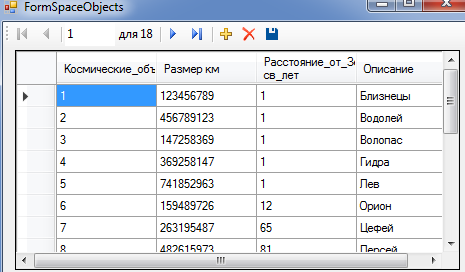


Рисунок 33 – Окно формы «Космические объекты»

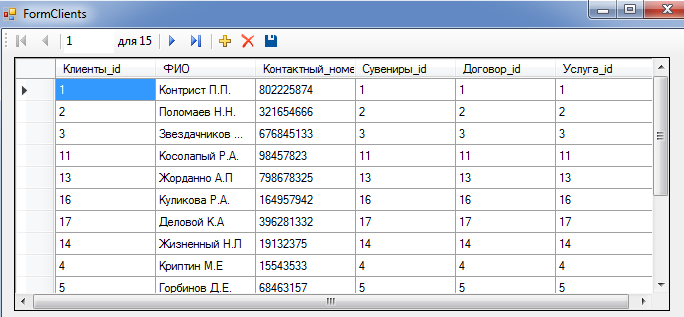


Рисунок 34 – Окно формы «Клиенты»



Рисунок 35– Окно формы «Должность»

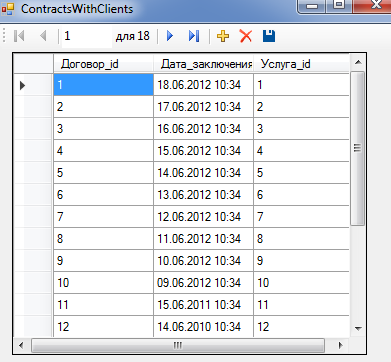


Рисунок 36 – Окно формы «Договор»

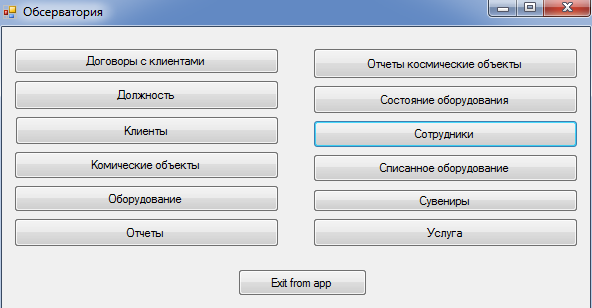


Рисунок 37 – Окно формы «Выбор таблицы»

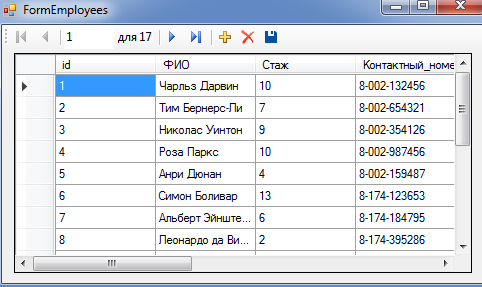


Рисунок 38 – Окно формы «Сотрудники»

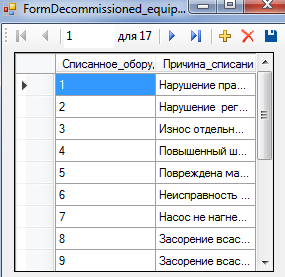


Рисунок 39 – Окно формы «Списанное оборудование»

# Результаты тестирования информационной системы

В результате тестирования приложения были выявлены ошибки при вставке, удалению, обновлению данных, которые были устранены. Были исправлены ошибки с авторизацией пользователей.

Таким образом, можно сделать вывод, что система полностью функционирует и готова к использованию.

В ходе тестирования в системе проверке подвергались:

* работа поисковой системы;
* загрузка формы при открытии таблицы «Клиенты»;
* работа экранных форм клиентского приложения;
* отказоустойчивость информационной системы;

# Заключение

В ходе разработки была спроектирована многопользовательская система для автоматизации сбора данных и обработки данных для Обсерватории солнечной динамики .

Данное приложение позволяет создавать необходимые документы, а также добавлять необходимые данные без значительных временных затрат.

Дублирование информации устранено.

Возможно дальнейшее расширение системы в случае необходимости. Благодаря объектно-ориентированному подходу к разработке и общей модульности системы можно быстро добавить необходимый функционал в приложение и создать необходимые функции в БД.

Таким образом, было получено удобное, надёжное и легко модифицируемое решение для управления интернет-магазином. В будущем планируется добавить Web-версию данного приложения.

При создании информационной системы использовались Visual Studio 2019, средство моделирования баз данных All Fusion Data Modeler 7, средство моделирования бизнес-процессов All Fusion Process Modeler 4.1, среда SQL Server Management Studio 18, язык баз данных T-SQL, ипользовалась технология ADO.NET, которая была интегрирована с современным языком программирования C#.

Разработанная информационная система может похвастаться легкостью в управлении, понятным интерфейсом и быстрой скоростью освоения самой программы, что в свою очередь делает ее доступной для работы людям с любым уровнем знания информационных систем.

# Список литературных источников

1. Горошкова Е. А. , Новиков Б. А. Основы технологий баз данных , 2018. – 240 с.
2. П. Вайнберг, Дж. Грофф, Эндрю Оппель. SQL. Полное руководство , 20019. – 960 с.
3. Мартин Грабер. SQL для простых смертных , 2009. – 672 с.
4. Линн Бейли. Изучаем SQL, 2012.
5. Филипп Андон, Валерий Резниченко. Язык запросов SQL. Учебный курс. 2008.
6. Дунаев В.В Базы данных. Язык SQL для студента. 2007.
7. Itzik Ben-Gan SQL Fundamentals for Microsoft SQL Server 2012 and SQL Azure. 2012.