

**Платформа TagNet.  
Модуль Конфигуратор (cf2)  
Руководство пользователя**

# ОГЛАВЛЕНИЕ

<b>НАЗНАЧЕНИЕ .....</b>	<b>3</b>
<b>МОДЕЛЬ ДАННЫХ .....</b>	<b>5</b>
Объектная модель данных .....	5
Типы объектов .....	5
Наследование типов .....	5
Параметры объектов .....	6
Группы свойств .....	8
Значение группы свойств.....	9
Элементы иерархии.....	9
Экземпляры объектов .....	9
Классы объектов.....	10
Связи между объектами.....	10
Теги     11	
<b>УСТАНОВКА И НАЧАЛО РАБОТЫ С КОНФИГУРАТОРОМ .....</b>	<b>13</b>
Установка Конфигуратора.....	13
Начало работы с конфигуратором .....	13
Резервное копирование базы данных .....	15
<i>Создание резервной копии базы данных.....</i>	<i>15</i>
<i>Восстановление базы из резервной копии.....</i>	<i>15</i>
<b>ОСНОВЫ РАБОТЫ С КОНФИГУРАТОРОМ .....</b>	<b>17</b>
Основные элементы интерфейса.....	17
Представление модели данных в приложении .....	17
Создание и удаление типа объекта .....	21
Создание и удаление элемента иерархии ОУ .....	22
Создание и удаление объекта.....	23
Редактирование параметров объекта.....	24
Редактирование групп свойств объекта/типа.....	25
Редактирование групп свойств параметра объекта/типа .....	26
Редактирование связей объекта .....	27
Генерация отчётов .....	27
Анализ модели .....	29

## ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Тег	Именованный параметр технологического объекта
Параметр объекта	
Объект управления	Технологический объект, обладающий набором измеряемых или контролируемых параметров. Может иметь составную структуру.

ТС	Телесигнализация
ТИ	Телеизмерение
ТИИ	Телеизмерение интегральное
ТУ	Телеуправление

# 1 Назначение

**Конфигуратор** (CF2) предназначен для описания объектной модели технологических объектов и процессов предприятия, и привязки объектной модели к модулям ввода/вывода платформы TagNet, и соответствующим протоколам.

Результатом работы конфигуратора являются конфигурационные файлы различных модулей платформа TagNet или для сторонних приложений.

Примеры получаемых из конфигуратора данных:

- конфигурационные файлы для модулей ввода/вывода платформы TagNet;
- конфигурационные файлы для SCADA-систем (списки тегов, привязка тегов к источникам данных и др.);
- отчеты по перечню и свойствам описанных в конфигураторе объектов.

Конфигуратор построен на основе плагинов, что дает широкие возможности по настройке его функциональности.

API плагинов Конфигуратора позволяет реализовать:

- новые модули для описания свойств объектов;
- новые отчеты для создания конфигурационных файлов;
- новые модули для взаимодействия со сторонними подсистемами;
- новые модули для изменения внешнего вида конфигуратора;
- подключения к различным СУБД для хранения введенной информации (PostgreSQL, HSQL и другие СУБД, поддерживающие JDBC драйвер).

## **2 Модель данных**

### **2.1 Объектная модель данных**

Представление данных в конфигураторе, их взаимосвязь и правила редактирования, называется моделью данных.

Модель данных в конфигураторе построена по объектному принципу, т.е. все технологические объекты управления (ОУ) описываются некоторыми объектами в модели.

Описание ОУ в модели проходит в три этапа. На первом этапе создается тип, определяющий структуру описания объекта. Затем создается элемент иерархии, соответствующий типу объекта. Элемент иерархии определяет место объектов данного типа в общей структуре ОУ. На третьем этапе создается экземпляр объекта данного типа, определяющий значения элементов описания конкретного объекта. Для одного типа может быть создано произвольное количество элементов иерархии и экземпляров объектов.

### **2.2 Типы объектов**

Тип объекта управления определяет структуру описания объектов управления одинакового предназначения, например, выключателей одной модели.

Для типа объекта заводятся набор свойств (параметров и состояний), которые характеризуют объект, и которые могут быть получены с датчиков и преобразователей непосредственно с объекта управления. Таким образом задаются общие для всех объектов управления данного типа свойства.

Для типа указываются:

- Наименование типа;
- Префикс для формирования имени тега, который используется для формирования имени тега.
- Текстовый описание.

Также для типа можно указать, будет ли он вообще участвовать в формировании имени тега.

### **2.3 Наследование типов**

Механизм наследования позволяет на основании одного типа – базового, построить другой – наследник, обладающий всеми свойствами родительского класса и определяющий дополнительный набор собственных свойств.

Механизм наследования предназначен для упрощения процесса моделирования за счёт объединения общих частей нескольких различных типов в единый родительский тип.

Примером использования механизма наследования может служить, например, родительский тип – **«Высоковольтный выключатель»** и тип-наследник – **«Высоковольтный выключатель с защитой»**.



На диаграмме показан пример:

- Базовый тип – **«Высоковольтный выключатель»**, в котором заведено 4 параметра (**Положение**, **Ток**, **Активная мощность**, **Реактивная мощность**).
- Тип наследник – **«Высоковольтный выключатель с защитой»**, наследует от отцовского типа 4 параметра и добавляет к ним еще один – **«Срабатывание защиты КЗ»**

## 2.4 Параметры объектов

Параметры объектов используются для описания сигналов, снимаемых с объекта и принимаемых объектом сигналов управления.

Для параметра определена следующая информация:

- Наименование параметра;

- **Тип параметра** – определяет тип значения измерения, снимаемого с объекта.

Возможные значения:

- **BOOLEAN** – булевское значение (истина/ложь);
- **INTEGER** – целочисленное значение;
- **REAL** – дробное значение;
- **STRING** – строковое значение;
- **Суффикс тега** – суффикс используется для формирования имени тега, соответствующего данному сигналу (уникально в пределах объекта);
- **Способ доставки (in/out)**
  - **in** - параметр объекта управления является входным, т.е. поступает на вход объекта управления. Например, это может быть сигнал телеуправления
  - **out** - параметр уходит из объекта управления, т.е. читается внешней системой
  - **in/out** - параметр является как входным, так и выходным.
  - **memory** - параметр является виртуальным. Не имеет отношения непосредственно к технологическому процессу и предназначен для уточнения представления объекта в системе.

Параметры определяются в типе объекта. Параметры сохраняются при наследовании, т.е. если параметр был определён в родительском типе, то в типе-наследнике он также будет определён. Атрибуты параметра можно редактировать только в том типе, в котором объявлен данный параметр – в типах-потомках параметр доступен только для чтения.

Экземпляр объекта наследует все параметры, описанные в типе.

Возможны ситуации, когда экземпляр объекта (или тип) не должны иметь параметр, который описан в типе объекта (или в родительском типе). В этом случае в типе или в экземпляре объекта можно *скрыть* параметры (объявить *неиспользуемыми*).

Скрытые параметры не учитываются при построении отчётов по данным модели. Неиспользуемые параметры не являются удалённым, при необходимости их можно вернуть в первоначальное состояние.

Обратите внимание на особенности наследования скрытых параметров:

- если параметр объявлен скрытым в некотором типе, то в типах-наследниках параметр не будет скрытым;

- если параметр объявлен скрытым в типе объекта, то в экземплярах объекта данного типа невозможно будет вернуть параметр в список используемых.

## 2.5 Группы свойств

Объекты и параметры в модели данных могут иметь свойства. Для описания свойств вводятся группы свойств. Каждое свойство представляет собой именованное, типизированное значение. Таким образом, группа свойств является набором именованных значений. Группа свойств, в свою очередь, также имеет наименование.

Например, группа свойств описывающая полный IP адрес может иметь следующий вид:

Имя группы свойств: «IP-адрес»

Перечень свойств:

- IP адрес – строковое свойство;
- TCP Порт – целочисленное свойство.

Значением группы свойств является совокупность значений всех её свойств. Например, значение указанной группы свойств может иметь вид:

- IP адрес – «127.0.0.1»;
- TCP Порт – 80.

Группы свойств бывают двух видов:

- Группы свойств технологических объектов – определяют свойства всего объекта;
- Группы свойств параметров объектов – определяют свойства отдельного параметра объекта.

Процесс добавления к объекту группы свойств называется привязыванием группы свойств к объекту (или к параметру объекта). На привязывание группы свойств к параметру может быть наложено ограничение, зависящее от конкретной группы свойств. Как правило, группа может быть привязана только к параметру определенного типа. На привязывание группы свойств к объекту ограничений не накладывается.

Группа свойств может быть задана как для типа, так и для экземпляра объекта. При этом способы работы с группами свойств не меняются.

Набор групп свойств модели фиксирован. Он определен разработчиками в коде конфигулятора. Но может быть расширен путем добавления плагинов с описанием новых групп свойств.



## 2.6 Значение группы свойств

После привязки группы свойств к параметру (объекту), в качестве значений свойств записываются значения по умолчанию, прописанные в описании группы свойств.

Тип объекта наследует все группы свойств, привязанные к параметрам в родительском типе. Тип объекта не наследует группы свойств объекта, привязанные к родительскому типу. Экземпляр типа наследует все группы свойств, привязанные к соответствующему типу (группы свойств объекта) и все группы свойств, привязанные к параметрам объекта.

### **Обратите внимание на особенности наследования групп свойств:**

- группы свойств, привязанные к типу, не наследуются в типах-потомках;
- группы свойств, привязанные к параметру типа, в типах-потомках не редактируются (нельзя изменить значения свойств).

В экземпляре типа могут быть привязаны дополнительные группы свойств, как к объекту, так и к параметрам. Значения свойств групп свойств, унаследованных от типа, могут быть переопределены в экземпляре типа в соответствии со свойствами конкретного объекта.

## 2.7 Элементы иерархии

На основе типов объектов управления строится дерево элементов иерархии. Дерево элементов иерархии определяет структуру расположения объектов управления. Элемент иерархии представляет собой тип объекта управления, помещенный на определенное место в дереве иерархии. Для одного типа можно создать неограниченное число элементов иерархии.

## 2.8 Экземпляры объектов

Экземпляр объекта управления описывает конкретный объект управления в системе. Объект управления всегда создается на основе некоторого типа, при этом экземпляр наследует все атрибуты типа, к которому он принадлежит.

Структура каждого конкретного объекта может быть расширена за счёт добавления к нему новых групп свойств (подробнее о группах свойств см. «Группы свойств»). Кроме того, для каждого объекта можно указать набор неиспользуемых параметров, тогда они не будут учитываться при формировании отчетов, хотя, формально, всё равно будут для него определены (подробнее о параметрах см. «Параметры объектов»).

При описании объекта управления указываются:

- наименование объекта управления;
- номер на технологической схеме (уникальный в пределах родительского объекта);
- описание объекта (не обязательно).

## 2.9 Классы объектов

Одно из назначений конфигулятора состоит в построении **выходных документов** (или конфигурационных файлов) по данным модели. Каждый выходной документ формируется с помощью **программного модуля** приложения, который строится на основании модели данных. Для каждого модуля важно, чтобы данные, им обрабатываемые, были определённым образом структурированы. Например, может требоваться, чтобы на входе ему задавался объект определённого «**типа**». Однако имя типа не подходит для использования в качестве критерия соответствия типа объекта «типу» требуемому модулю. Это связано с тем, что типы объявляются динамически. Более того, можно создать два идентичных «типа», которые будут иметь различные имена и т.д.

В качестве **идентификатора «типа»** объекта используются специальные группы свойств объекта. Так как группы свойств определяются в коде приложения (статически), то никакой путаницы для отчётов, в понимании принадлежности объекта к тому или иному «типу», не возникает.

Группы свойств объекта определяют его принадлежность к некоторому *классу* объектов. Формально, любая группа свойств *A* определяет принадлежность объекта к классу объектов *A*. Фактически, определение группой свойств класса объектов зависит от способа использования данной группы свойств отчётами.

## 2.10 Связи между объектами

В реальной структуре объекты могут быть связаны между собой не только иерархическими связями. Может потребоваться связать между собой объекты из параллельных иерархий. Например, можно построить иерархию физических объектов и параллельно ей иерархию программных компонентов (сервера сбора данных, АРМы диспетчеров и т.д.). В этом случае необходимо указать, с каких объектов будет собираться данные конкретный сервер или какие объекты будут контролироваться со SCADA.

Для описания связей, отличных от иерархических, в конфигуляторе используются связи между объектами.

Связь представляет собой типизированный объект. Типы связей описываются статически в коде конфигулятора (аналогично группам свойств). Набор типов связей может быть расширен, путем добавления плагинов с описанием новых типов связей.

Тип связи имеет имя, которое характеризует предназначение связи. Каждая связь имеет направление, т.е. имеется объект начала связи и объект конца связи.

Связи могут быть протянуты между двумя объектами, объектом и параметром или между двумя параметрами.

Каждый тип связи накладывает ограничения на объект (параметр) начала связи и объект (параметр) конца связи. Ограничения касаются класса объекта (т.е. к объекту/параметру должна быть привязана группа свойств определенного типа).

Отдельно следует выделить связь между двумя параметрами – эта связь может иметь собственные свойства.

## 2.11 Теги

В АСУТП используется понятие **тега** как основного информационного элемента. Тег имеет уникальное имя и набор атрибутов. Тег является источником информации и может быть, например, сигналом с аналогового или дискретного датчика, данными, переданными из другого приложения, либо может быть введен пользователем.

Чтобы упростить формирование имен тегов и обеспечить уникальность имени тега, в конфигулятор встроен механизм формирования имени тега с использованием иерархической структуры данных.

Имя тега состоит из двух частей:

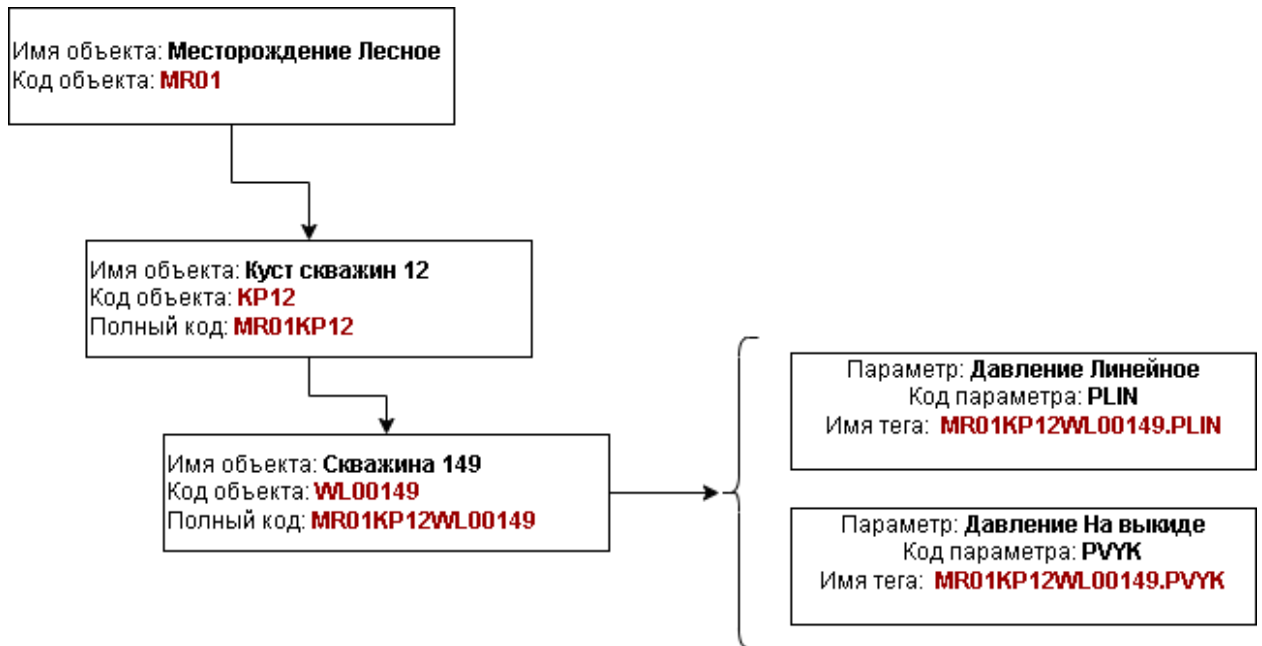
- **префикс** – часть имени тега, идентифицирует объект, которому принадлежит параметр;
- **суффикс** – часть имени тега, идентифицирует тип параметра.

Префикс имени тега обычно складывается из кодов всех объектов в иерархии, от родительского по отношению к текущему объекту. Код каждого объекта складывается из префикса, определенного для типа объекта и номер объекта на технологической схеме. По префиксу имени тега можно определить положение объекта в общей иерархии объектов управления.

Суффикс тега определяется в атрибутах параметра объекта, по суффиксу тега можно определить, какому типу параметра он соответствует.

Таким образом, имя тега полностью идентифицирует объект, и его параметр.

Уникальность имени тега обеспечивается за счет уникальности префиксов типов объектов и уникальности номеров объектов на тех. схеме в пределах родительского объекта. Подробнее о проверке уникальности имени тега см. «Анализ модели».



## 3 Начало работы с конфигуратором

### 3.1 Запуск Конфигуратора

Для работы программы требуется наличие установленной Java версии 8.

Убедитесь, что Java установлена на машине и путь к исполняемому файлу Java добавлен в переменную PATH.

Дистрибутив представляет архивный файл, который необходимо разархивировать в локальную папку, например, в папку:

```
c:\tagnet\cf2
```

Запуск программы осуществляется с помощью командного файла:

```
cf2.bat
```

или

```
cf2.sh
```

из папки программы.

### 3.2 Начало работы с конфигуратором

Для хранения данных конфигуратор использует базу данных. Конфигуратор может подключаться к различным СУБД в зависимости от выбранных настроек.

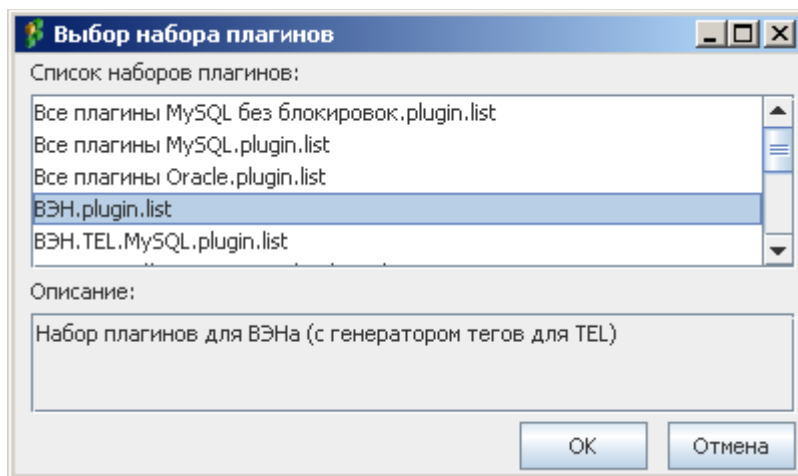
Для первого запуска конфигуратору созданная понадобится пустая база данных (схема). При подключении к пустой схеме, конфигуратор автоматически создаст все необходимые таблицы.

Если база данных не пуста и не содержит таблиц, относящихся к программе, запуск завершится с ошибкой.

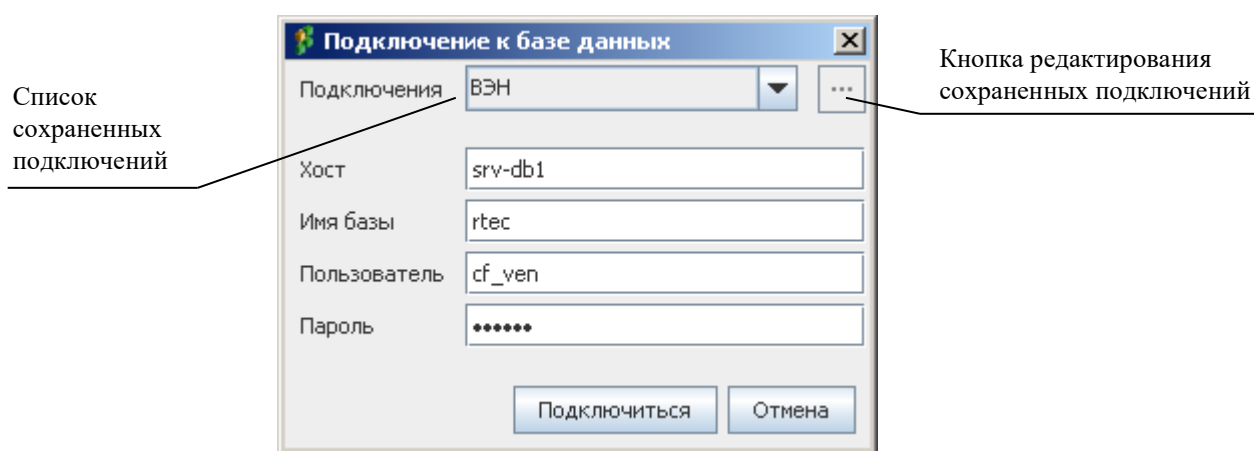
Если заполненная база данных поставляется в виде файла резервной копии, перед началом работы необходимо восстановить базу данных (см. 3.3.2 Восстановление базы из резервной копии).

При запуске конфигулятора открывается окно выбора списка набора плагинов.

Набор плагинов определяет, к какой базе данных будет подключен конфигуратор, работа с какими подсистемами будет доступна и т.д. Если список плагинов только один, он загружается автоматически, без открытия окна выбора списка плагинов.

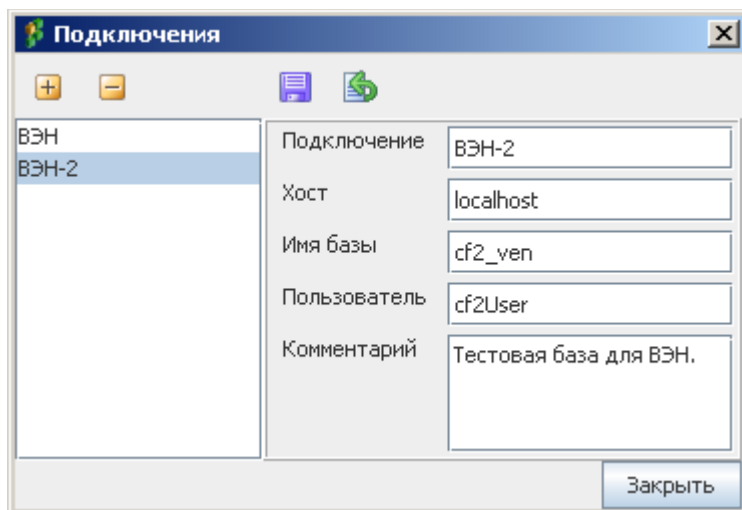


После загрузки выбранного списка плагинов, открывается главное окно конфигуратора и появляется диалог подключения к базе данных.



Вы можете выбрать одно из сохраняемых подключений из выпадающего списка «Подключение», либо ввести параметры подключения вручную, непосредственно в поля. Параметры последнего успешного подключения к базе данных (кроме пароля) сохраняются между запусками Конфигуратора.

Набор сохраняемых параметров подключений доступный в выпадающем списке «Подключение» предназначен для тех, кто работает с несколькими базами данных конфигуратора и вынужден каждый раз вручную вводить параметры подключений. Вы можете сохранить несколько именованных наборов параметров подключений и выбирать нужный набор из списка. На рисунке приведен вид окна редактирования набора сохраняемых подключений.



### 3.3 Резервное копирование базы данных

В дополнение к средствам резервирования информации, которые предоставляют СУБД, в конфигураторе реализован механизм создания резервной копии всех объектов базы данных и восстановления объектов БД из резервной копии. В отличие от использования механизмов СУБД, резервная копия, созданная из конфигуратора, может быть развернута на другой СУБД. Например, конфигуратор работает с базой данных MySQL. Чтобы перевести базу данных на другую (PostgreSQL или MS SQL), достаточно из конфигуратора сделать резервную копию базы данных MySQL, перезапустить конфигуратор, подключиться к новой базе данных и восстановить информацию из резервной копии.

#### 3.3.1 Создание резервной копии базы данных

Создание резервной копии выполняется после подключения к базе данных.

Для создания резервной копии, необходимо выбрать пункт главного меню «Управление -> Сделать резервную копию БД...». В появившемся диалоге, надо указать файл, в который будет сохранена копия, и нажать кнопку «ОК». Начнется процесс создания резервной копии БД.

#### 3.3.2 Восстановление базы из резервной копии

Восстановление из резервной копии выполняется после подключения к базе данных конфигуратора.

Для восстановления базы данных из резервной копии, необходимо выбрать пункт главного меню «Подключение -> Восстановить из резервной копии БД...».

В диалоге, необходимо указать файл резервной копии и нажать кнопку «ОК».

Затем появляются диалоги подтверждения восстановления, которые предназначены защитить базу данных от непреднамеренного удаления в ходе восстановления из резервной копии.



**Предупреждение:** Обратите внимание, при восстановлении из резервной копии, база данных будет очищена и все данные будут потеряны.

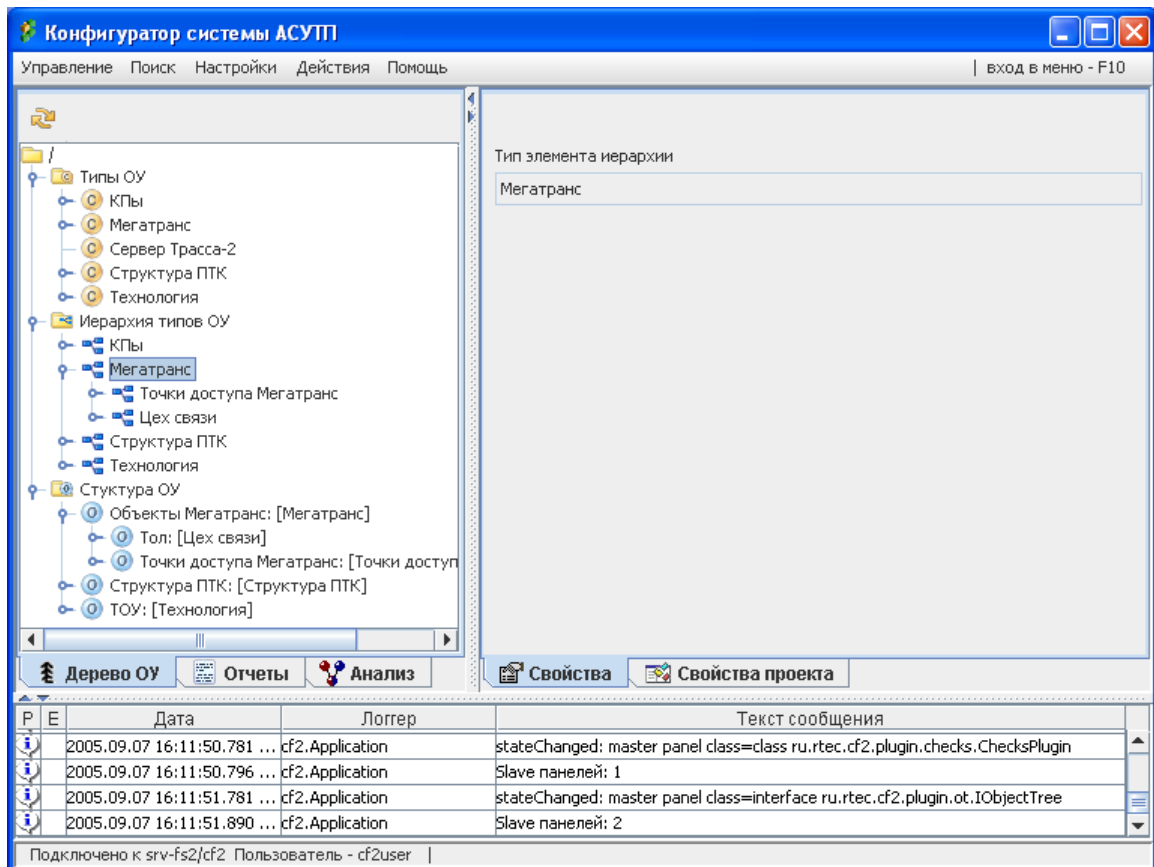


## 4 Основы работы с конфигуратором

### 4.1 Основные элементы интерфейса

Интерфейс приложения состоит из следующих основных компонентов.

- Главное меню;
- Панель выбора объекта (в левой части окна);
- Панель редактирования свойств объекта/параметра (в правой части окна);
- Панель отображения отладочной информации (в нижней части окна).



Панель выбора содержит следующие вкладки:

- Дерево ОУ – отображает состояние текущей модели данных;
- Отчёты – отображает список отчётов, которые можно построить по данным модели;
- Анализ – отображает список проверок, которые можно произвести над моделью.

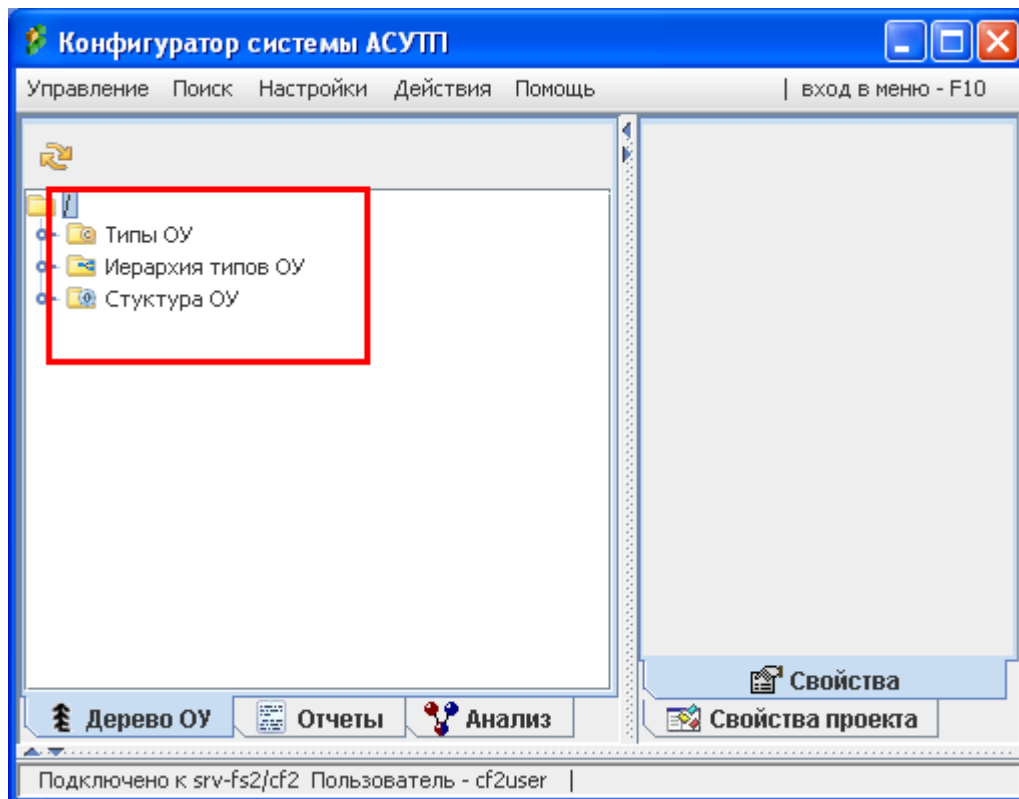
### 4.2 Представление модели данных в приложении

Модель данных в приложении построена на объектном принципе. Каждому объекту управления (ОУ) соответствует в модели:

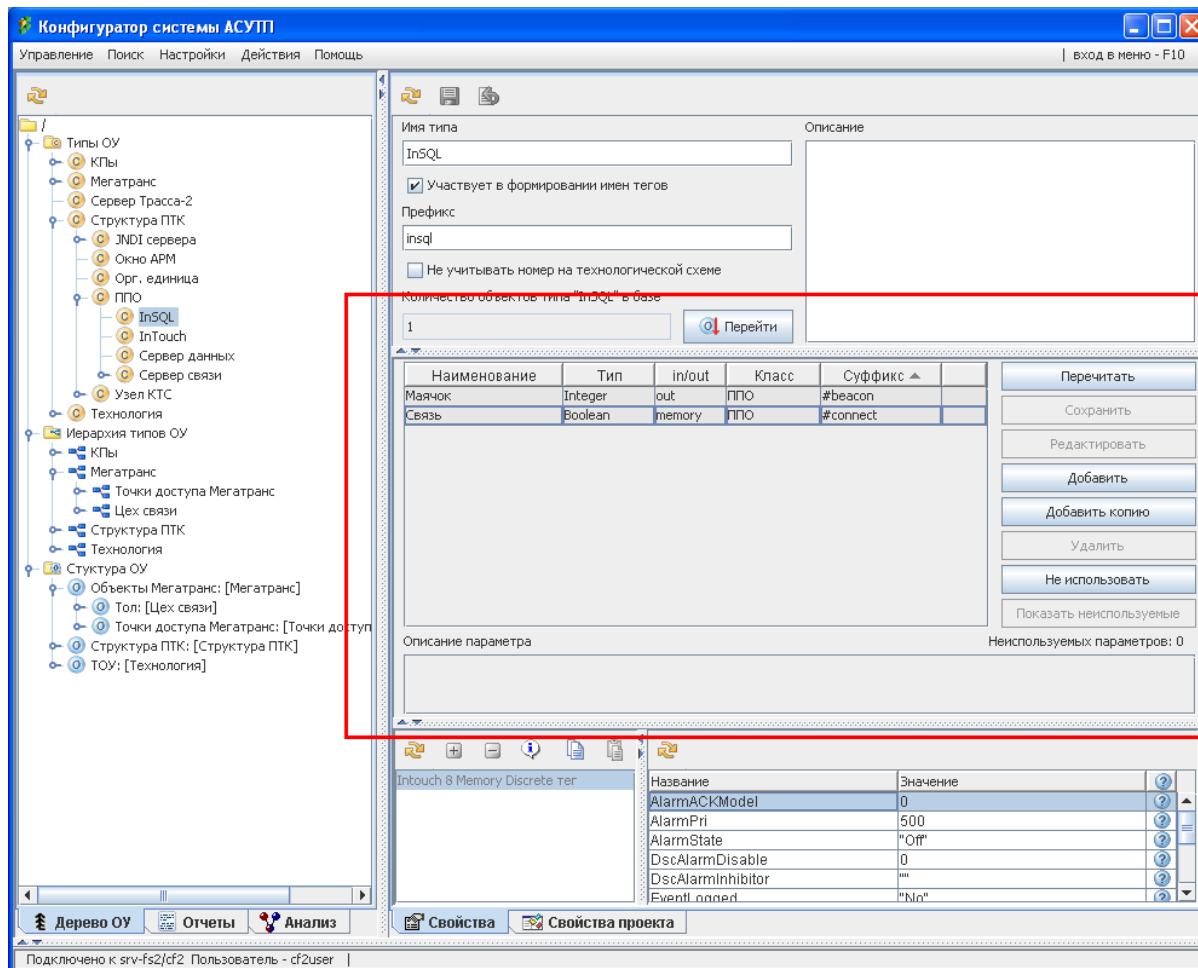
- типы объекта,

- иерархия типов,
- структура объектов

Все эти элементы модели данных отображаются в дереве навигации конфигуратора.



Тип объекта управления задаёт описание параметров, которыми обладает каждый объект данного типа. Типы объектов имеют иерархическую структуру, и на базе одного типа может быть построен другой тип – наследник, который включает все параметры родительского типа и, кроме того, может определять свои собственные параметры.



Параметр имеет следующие характеристики:

- наименование (строка);
- тип значения параметра;
  - String – строка;
  - Boolean – логическое значение (истина/ложь);
  - Integer – целое число;
  - Real – вещественное число;
- свойство in/out: (in, out, in-out, memory)
  - in – параметр поступает на вход объекта управления, например, сигнал телеуправления;
  - out – параметр уходит из объекта, т.е. читается внешней системой;
  - in-out – параметр является как входным так и выходным;
  - memory – параметр является виртуальным, т.е. не имеет отношения к технологическому процессу, а предназначен для уточнения представления объекта в системе.
- класс – имя класса, в котором определён данный параметр, данное поле не для редактирования;

– суффикс (строка) – суффикс, который будет добавляться к именам тегов, связанных с этим параметром.

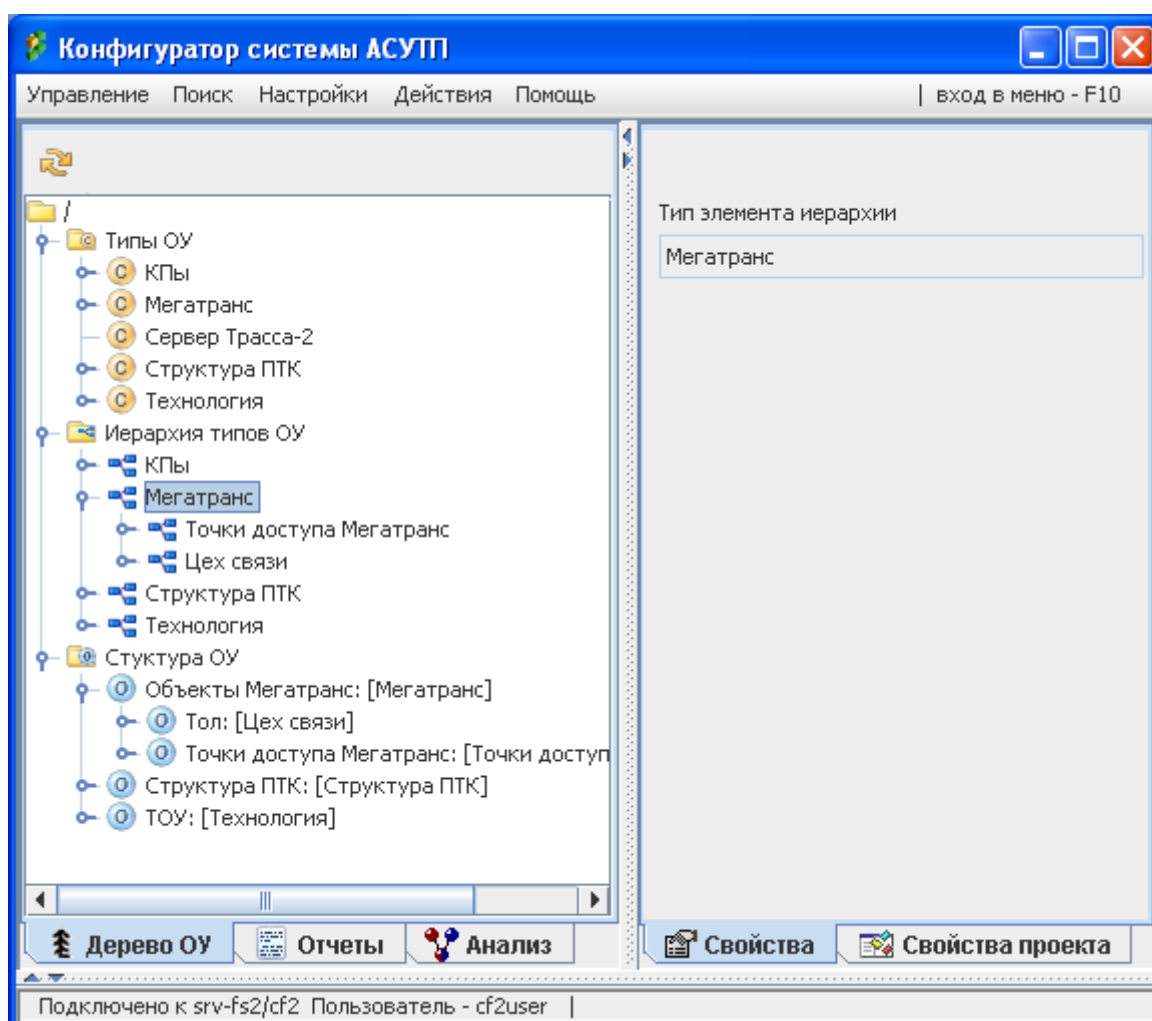
С каждым параметром объекта может быть связано несколько групп свойств. Каждая группа свойств имеет тип. Группа свойств представляет собой набор именованных типизованных полей со значениями по умолчанию. При создании экземпляра объекта данного класса, его группы свойств будут иметь указанные значения.

Группы свойств объекта используются для указания принадлежности данного объекта к некоторому *классу* объектов. Так все объекты, помеченные группой свойств «*OPC DA Сервер*», считаются серверами OPC DA.

Группы свойств предназначены для указания значений свойств параметра или объекта, необходимых для построения по нему того или иного отчёта.

Структура ОУ определяет иерархию экземпляров объектов. Каждый объект, входящий в состав структуры ОУ, имеет определённый тип, который описывается в типах ОУ. Соответственно, элемент структуры ОУ описывает свойства некоторого технологического объекта или оборудования.

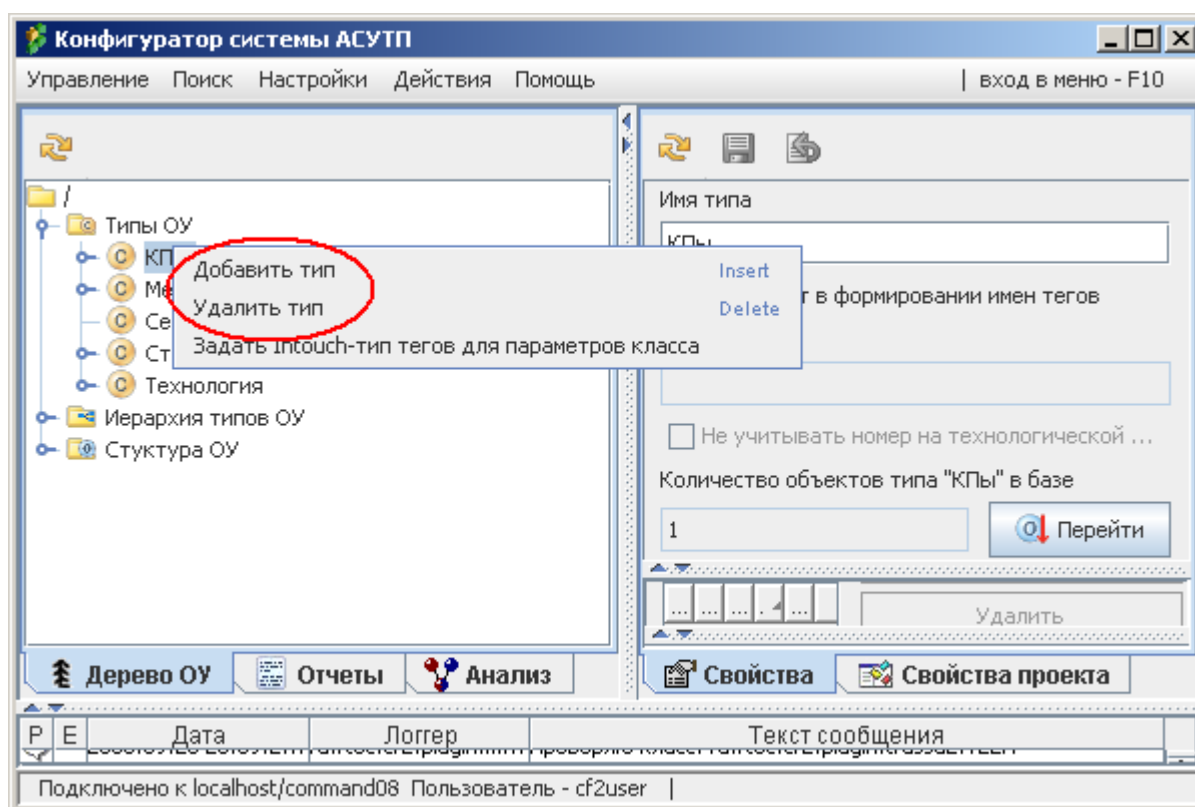
Иерархия типов ОУ определяет иерархическую структуру ОУ.



Так, например, можно указать, что в корне иерархии структуры ОУ может располагаться объект типа «Мегатранс» в состав, которого могут входить *только* объекты типа «Точка доступа Мегатранс» и типа «Цех связи», причём их количество может быть произвольным, в т.ч. тип может быть вообще не представлением ни каким объектом. Соответственно, в данном случае в структуре ОУ имеется объект «Объект Мегатранс» типа «Мегатранс», в состав которого входят объекты «Точки доступа Мегатранс» типа «Точки доступа Мегатранс» и «Тол» типа «Цех связи». В свою очередь каждый из вложенных объектов может иметь свои вложенные объекты и т.д.

Объекты могут быть соединены между собой связями определённого типа. Тип связи определяет ограничения на классы объектов, которые она может соединять.

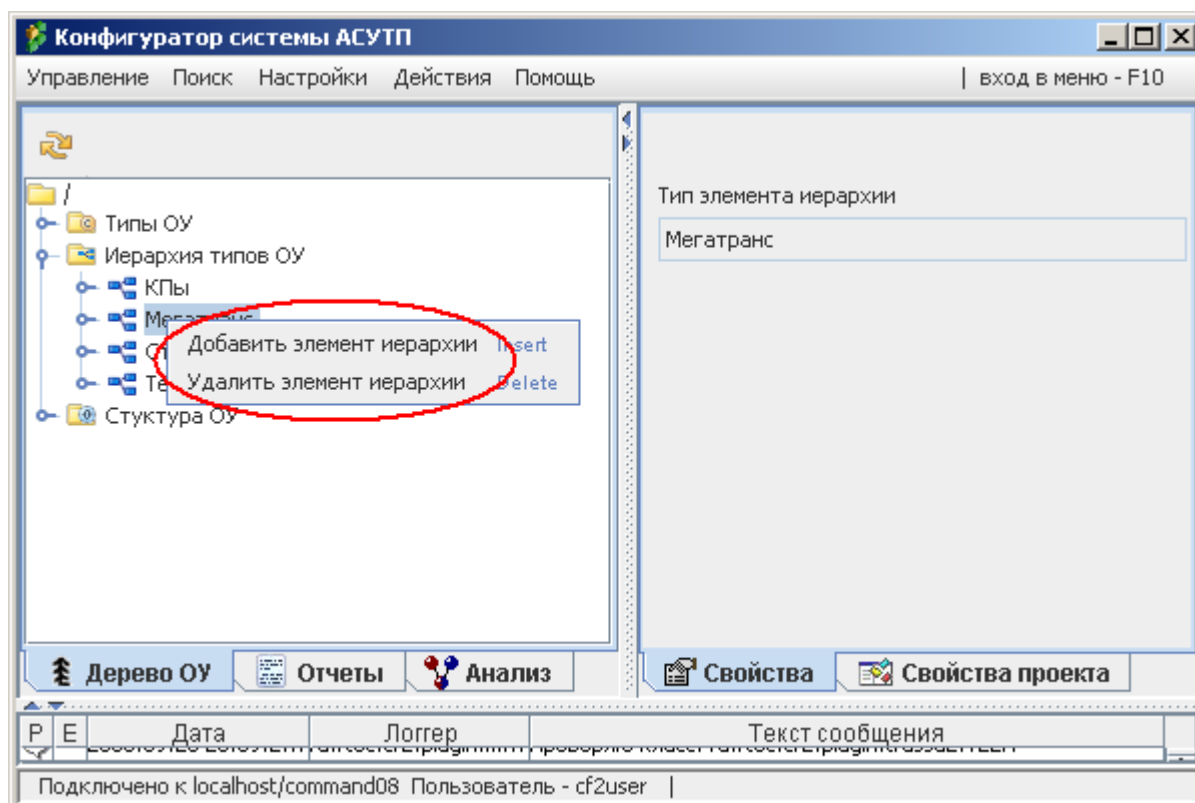
### 4.3 Создание и удаление типа объекта



Для создания/удаления типа объекта необходимо кликнуть правой кнопкой мыши по элементу дерева типов ОУ и выбрать пункт меню «Добавит тип» или «Удалить тип». При этом если создавать тип от корня дерева, то он не будет иметь родительского типа. Если создавать тип от другого типа, то тот тип будет считаться для данного родительским, и созданный объект будет находиться в дереве типов ОУ «внутри» родительского типа.

Удалить тип можно только в том случае если он не имеет наследников и экземпляров.

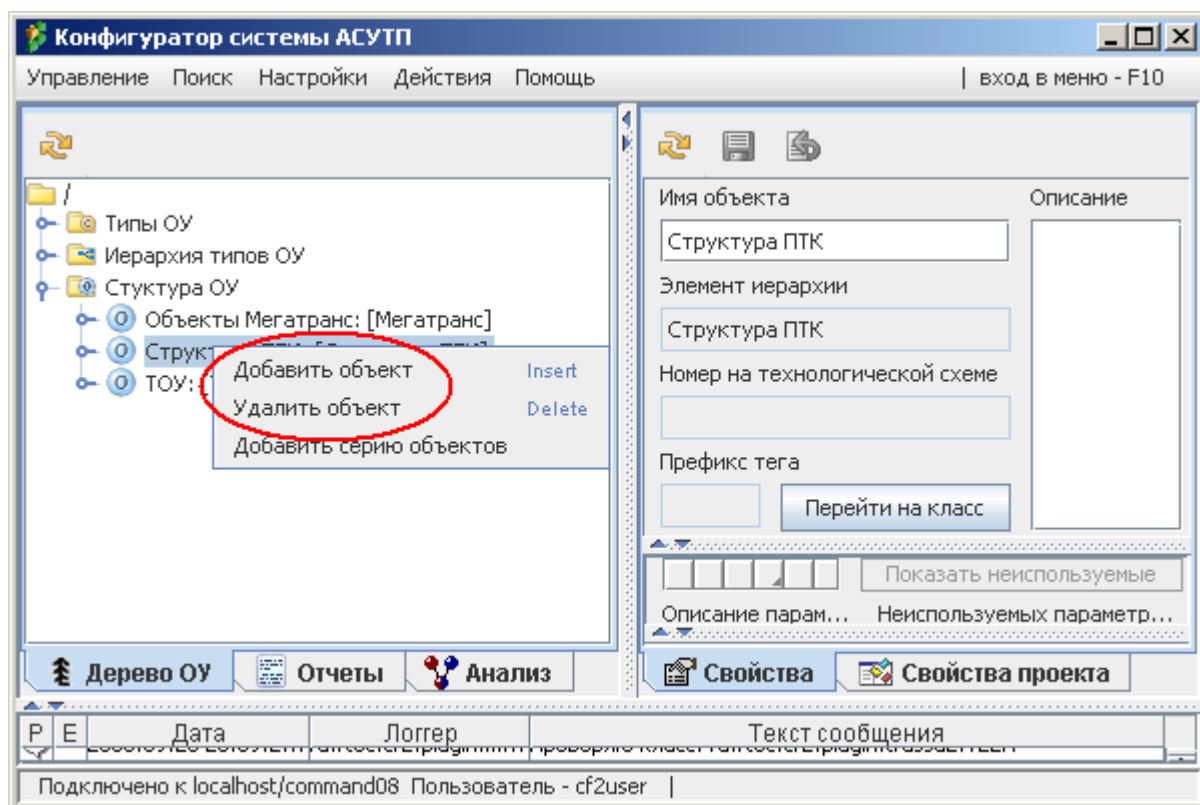
## 4.4 Создание и удаление элемента иерархии ОУ



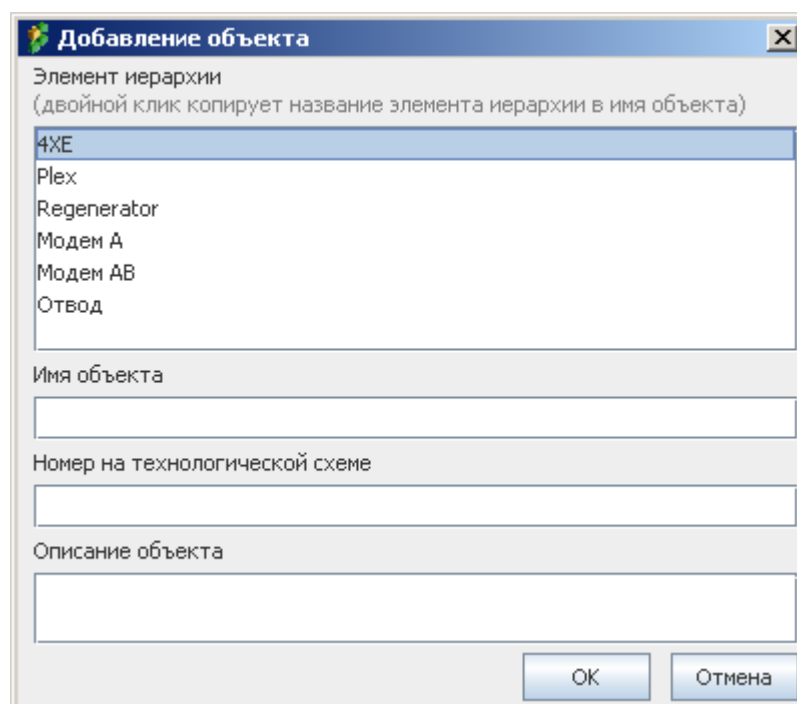
Для добавления (удаления) элемента иерархии типов объектов необходимо кликнуть правой кнопкой мыши на элемент иерархии, в который требуется поместить новый тип (который требуется удалить), после чего выбрать добавить элемент иерархии (удалить элемент иерархии).

Удалить элемент иерархии можно в том случае если он не имеет вложенных элементов, и в структуре объектов не определено объекта в позиции определяемой данным элементом иерархии.

## 4.5 Создание и удаление объекта

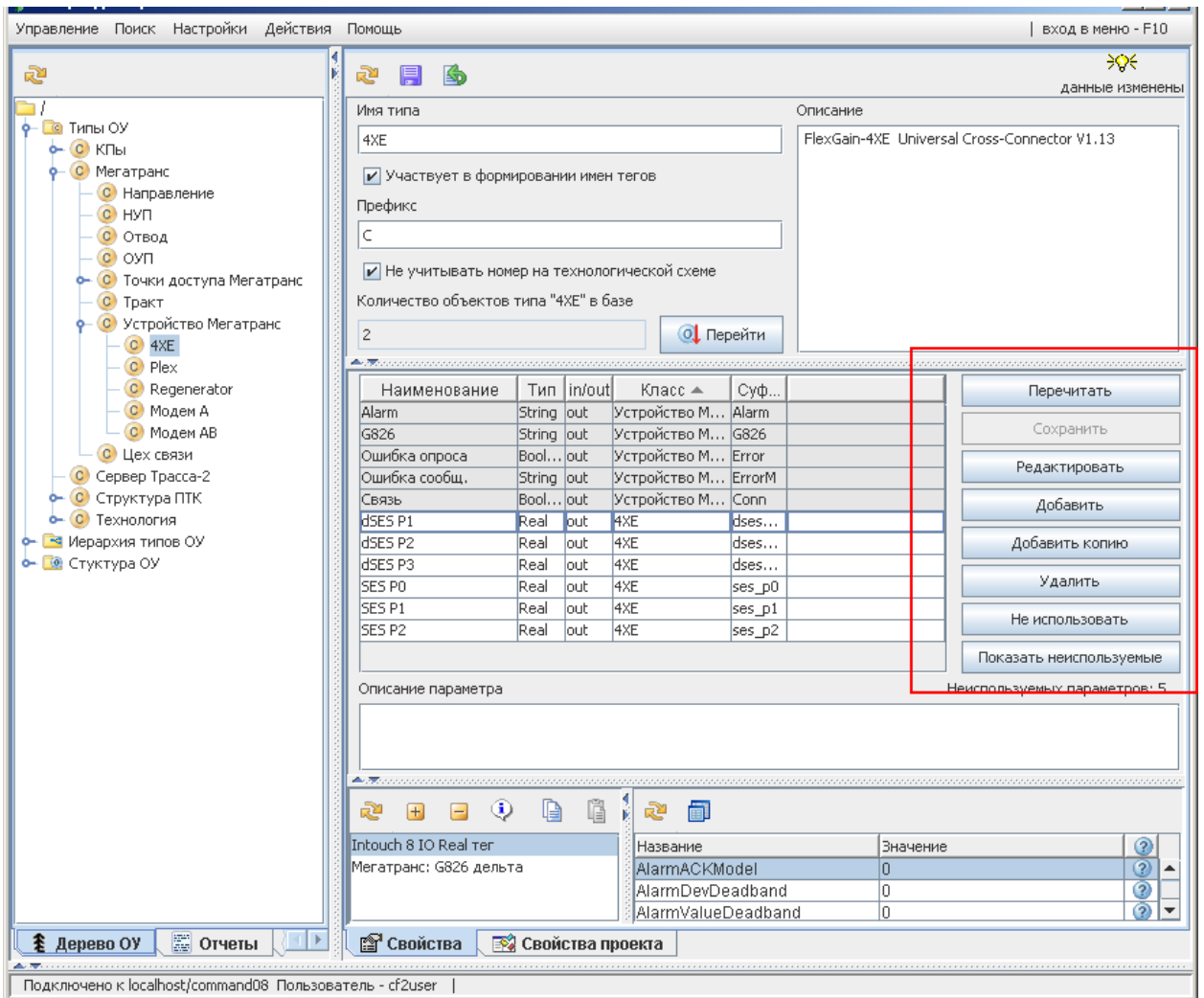


Создание/удаление объекта аналогично созданию/удалению типа, только действия необходимо производить в структуре объектов. При добавлении объекта необходимо выбрать его тип:



Тип объекта необходимо выбрать из списка, который определяется иерархией типов объектов.

## 4.6 Редактирование параметров объекта

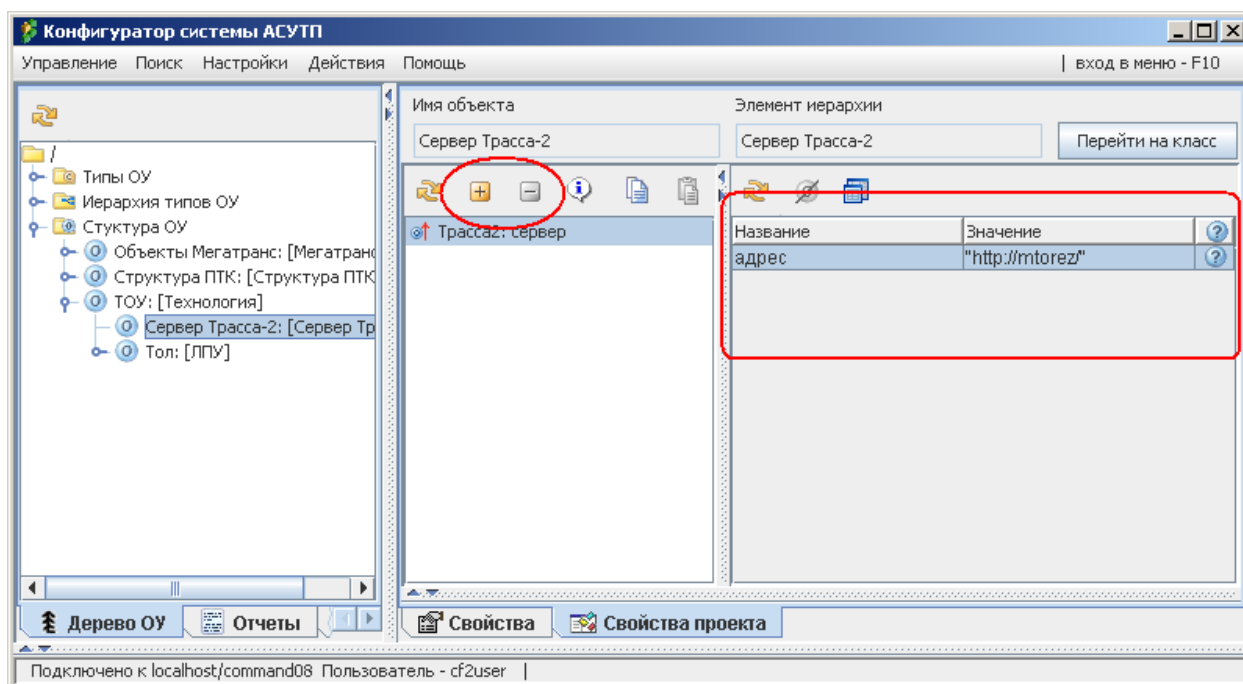


Для редактирования набора параметров типа, необходимо выбрать данный тип в иерархии типов ОУ.

- Для добавления параметра используется кнопка «Добавить»;
- Для удаления параметра – «Удалить»;
- Для редактирования параметра – «Редактировать»;
- Для того чтобы указать параметр как неиспользуемый – «Не использовать»;
- Для того чтобы сделать параметр вновь используемым – «Использовать» (появляется на месте «Не использовать» при выделении неиспользуемого параметра);
- Все неиспользуемые параметры по умолчанию не отображаются, для их отображения используется кнопка «Показать неиспользуемые».



## 4.7 Редактирование групп свойств объекта/типа

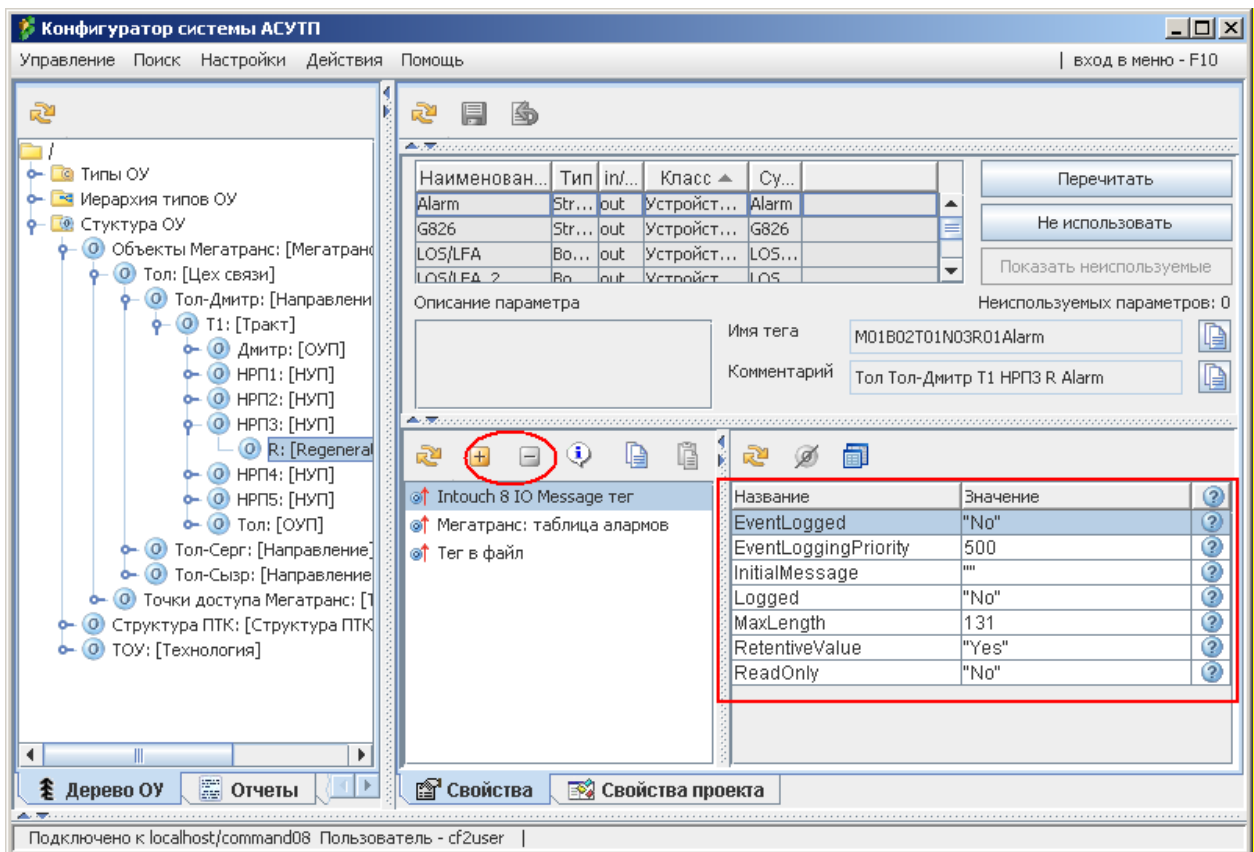


Редактирование групп свойств объекта (типа) происходит во вкладке «Свойства проекта». Для добавления/удаления группы свойств используются кнопки «плюс»/«минус», выделенные на рисунке овалом.

Редактирование группы свойств происходит в панели выделенной прямоугольником. В ней отображается набор имён свойств и соответствующих им значений группы свойств. Для редактирования значение свойства необходимо дважды по нему кликнуть.

Для объекта группы свойств определённые в типе помечаются значком – .

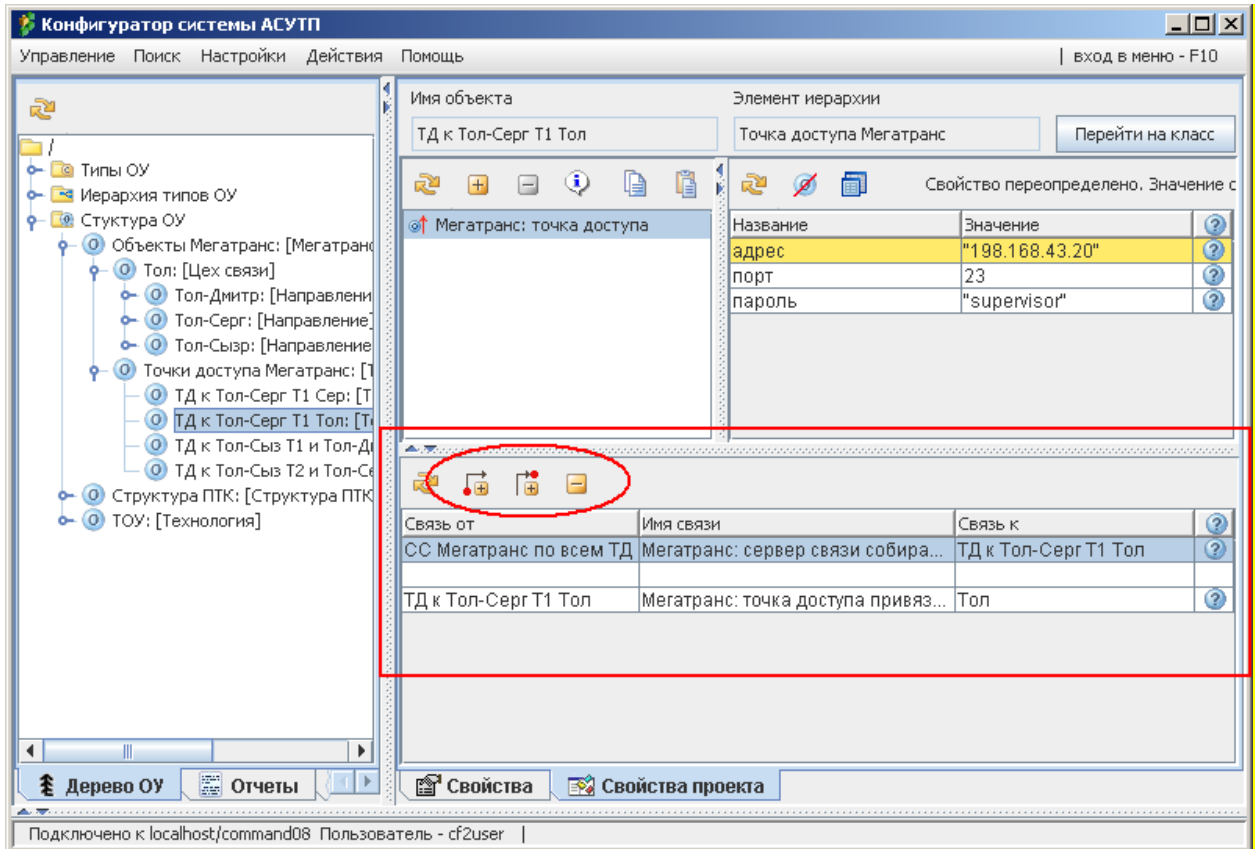
## 4.8 Редактирование групп свойств параметра объекта/типа



Для добавления группы свойств к параметру объекта/типа необходимо выделить соответствующий параметр и нажать на кнопку «Плюс» (на рисунке выделено овалом).




Для редактирования свойств группы свойств параметра используется панель, выделенная на рисунке прямоугольником. Для изменения значения отдельного свойства необходимо дважды щёлкнуть по нему мышью и далее изменить его значение в окне редактирования.

## 4.9 Редактирование связей объекта



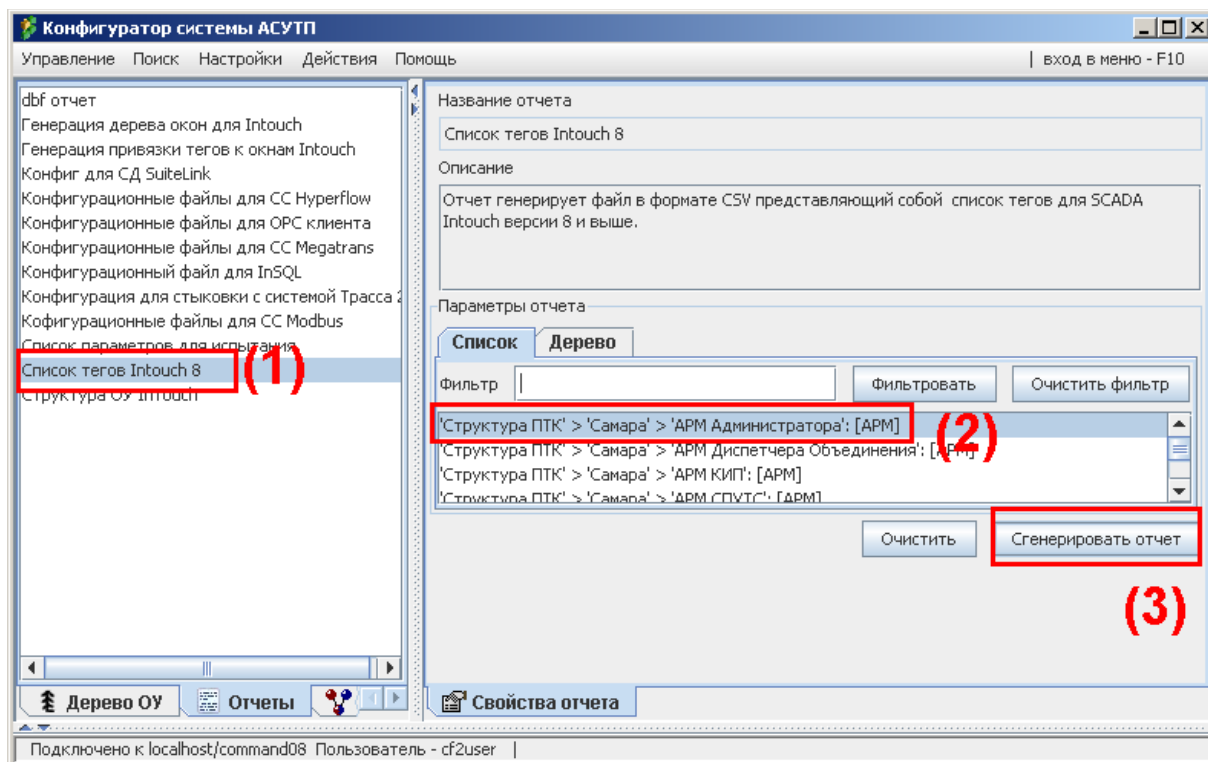
Для редактирования связей объекта используется панель в свойствах проекта, выделенная на рисунке прямоугольником.

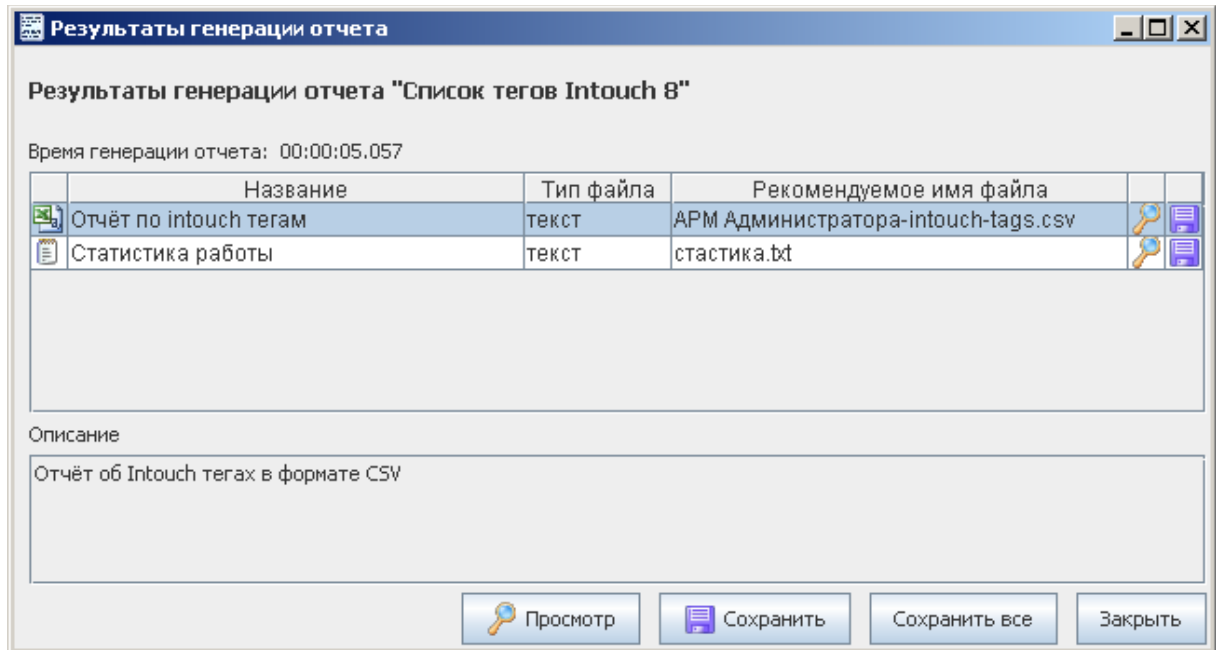
Для редактирования связей используются кнопки, обведённые на рисунке в овал.

-  – для добавления связи от данного объекта к другому объекту;
-  – для добавления связи от другого объекта к данному;
-  – для удаления связи.

## 4.10 Генерация отчётов

Каждый отчёт представляет собой некоторый набор файлов, содержимое которых сформировано на основании данных модели. Для генерации отчёта необходимо на вкладке «Отчёты» выбрать требуемый тип отчёта (1), указать необходимые для отчёта параметры (2) и сгенерировать отчёт (3).

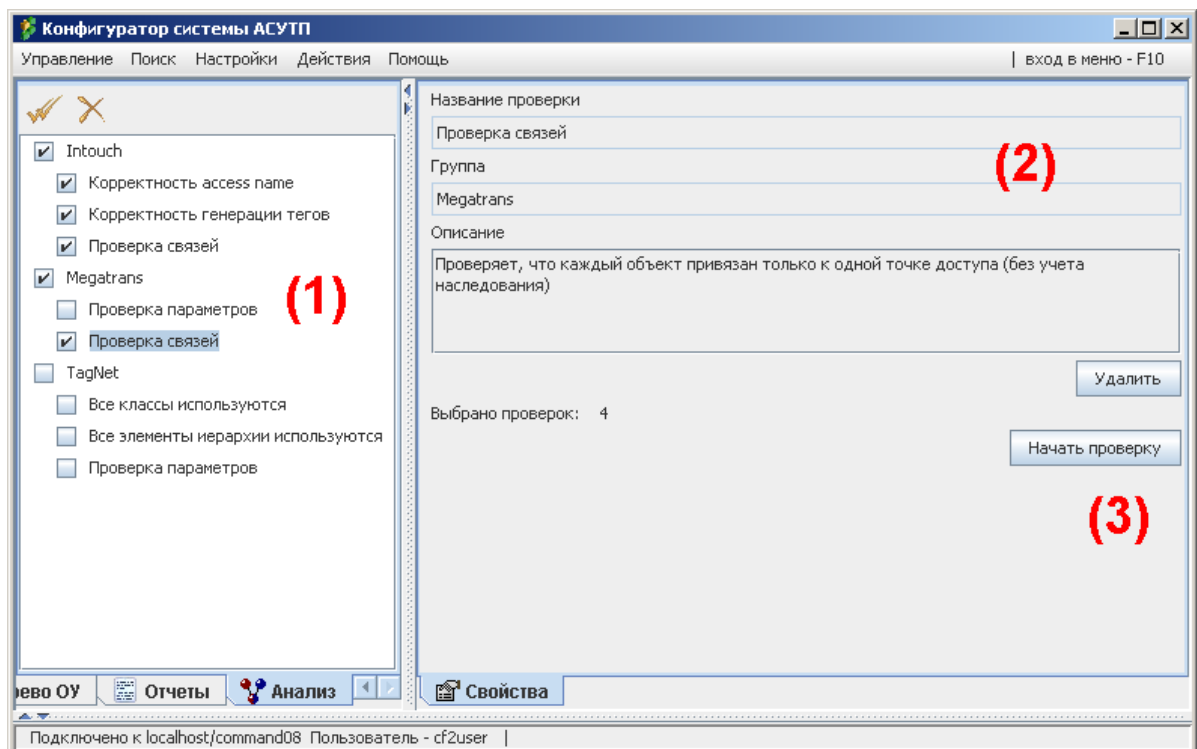




Полученные файлы можно сохранить.

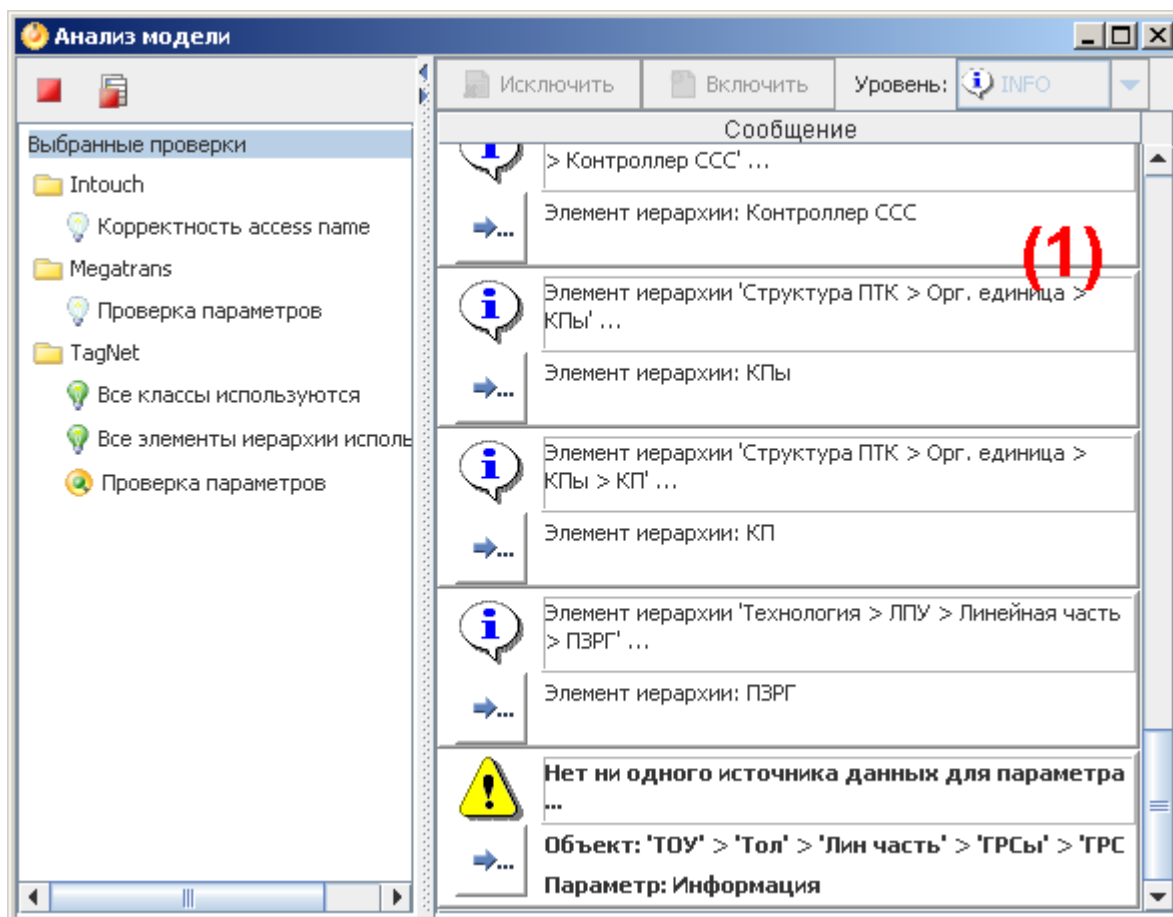
## 4.11 Анализ модели

Для анализа модели используются проверки. Все проверки разбиты по категориям в зависимости от их назначения. Так проверки входящие в категорию Megatrans проверяют корректность структуры объектов Megatrans.



Пользовательский интерфейс анализа модели имеет следующую структуру (см. рисунок выше):


1. Список имеющихся проверок, позволяет включить либо выключить исполнение проверок.
2. Описание проверки.
3. Кнопка *начать проверку* запускает на исполнение включенные проверки. Проверки исполняются последовательно, одна за другой.



Результатом работы проверок является список сообщений (событий) о найденных несоответствиях в модели, (1) на рисунке. Каждое событие имеет статус, который указывает на степень критичности события:

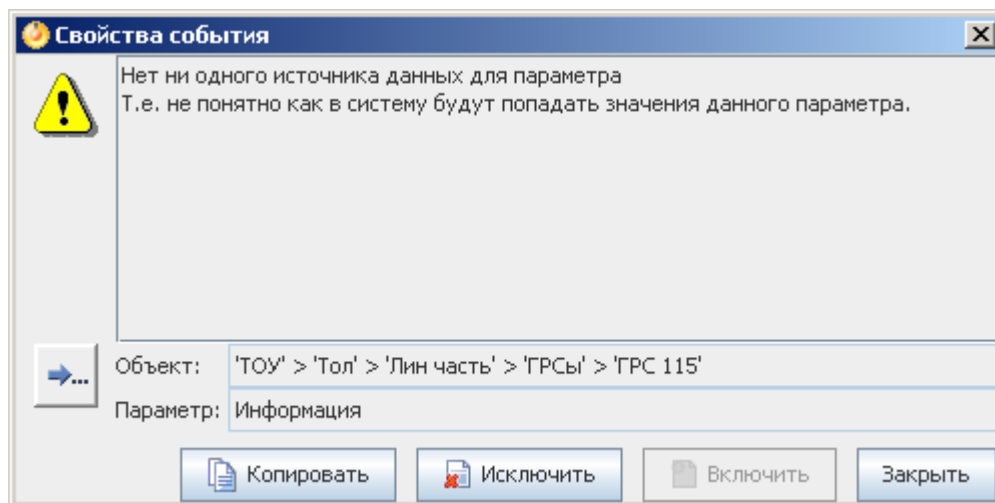
- INFO – информационное – несоответствие не влияет на работу конфигуратора (например, сообщение о неиспользуемом классе или элементе иерархии);
- WARNING – предупреждение – возможно, неправильно описаны свойства объектов и построена иерархии;
- ERROR – ошибка – ошибка в описании объектов, которая должна быть исправлена, например, найдены теги с дублирующимися именами.

Как правило, каждое событие относится к некоторому объекту, элементу иерархии, типу или параметру объекта. Информация об объекте, к которому относится событие, можно видеть как в таблице событий, так и в диалоге свойств событий. Для быстрого

перехода на объект события предназначена кнопка  «Переход на объект», которая переводит фокус в дереве ОУ на объект события.

Для получения подробной информации о сообщении необходимо дважды щёлкнуть на нём мышью.

Информационный диалог о событии имеет следующий вид:



Назначение кнопок диалога свойства события:

- «Копировать» – копировать текстовое описание события в буфер обмена;
- «Исключить» – пометить событие как обработанное;
- «Включить» – снять пометку о том, что событие обработано;
- «Закрыть» – закрыть диалог.

Возможность пометить обработанные события введена для упрощения процесса исправления модели, поскольку не всегда найденные ошибки удобно исправлять в порядке их появления. События, помеченные как обработанные, выводятся в перечеркнутом виде. По умолчанию, обработанные события больше не показываются в таблице, но их отображение можно включить.