

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

**Решение проектных задач  
с помощью онтологических систем**

Электронные методические указания  
к лабораторным работам

САМАРА

2010

УДК 004.9(075)  
ББК 32.97

**Составители: Боргест Николай Михайлович,  
Симонова Елена Витальевна,  
Шустова Дина Владимировна**

Методические указания к лабораторным работам по дисциплине «Онтология проектирования» являются частью контента образовательных программ магистратуры, разрабатываемого на основе использования новых образовательных технологий, ресурсов и систем электронного и дистанционного обучения для магистерской программы «Проектирование, конструкция и CALS-технологии в авиационной технике» по направлению 160100.68 «Авиастроение».

Указания разработаны на кафедре конструкции и проектирования летательных аппаратов в рамках разрабатываемого научного направления «Онтология проектирования».

© Самарский государственный  
аэрокосмический университет, 2010

## СОДЕРЖАНИЕ

Предисловие.....	8
ЛАБОРАТОРНАЯ РАБОТА №1 .....	10
1.1. Часть 1. Выбор транспортного средства .....	10
Введение.....	10
1.1.1. Цели и задачи лабораторной работы №1 .....	11
1.1.2. Создание онтологии.....	11
1.1.2.1. Создание проекта .....	11
1.1.2.2. Сохранение проекта.....	12
1.1.2.3. Создание классов .....	13
1.1.2.4. Создание класса «авиатранспорт».....	14
1.1.2.5. Создание классов «Ж/д транспорт», «Автомобильный транспорт» и «Речной транспорт».....	14
1.1.2.6. Редактирование дерева иерархии .....	15
1.1.2.7. Создание слотов .....	16
1.1.2.8. Включение слотов в класс.....	18
1.1.2.9. Связывание слота с классом .....	20
1.1.2.10. Создание отношения между классами.....	22
1.1.2.11. Настройка формы ввода экземпляров.....	23
1.1.2.12. Изменение параметров виджета .....	24
1.1.2.13. Создание экземпляров класса .....	30
1.1.2.14. Установка слота отображения .....	32
1.1.2.15. Изменение параметров виджета для создания семантического представления онтологии.....	32
1.1.2.16. Создание семантического представления онтологии.....	37
1.1.2.17. Создание запроса .....	41
1.1.2.18. Сохранение запроса .....	46
1.1.2.19. Загрузка запроса.....	47
1.1.3. Задания для самостоятельной работы магистрантов.....	47
1.2. Часть 2. Выбор самолета.....	49
1.2.1. Расширение онтологии .....	49
1.2.1.1 Загрузка проекта .....	49
1.2.1.2 Создание дополнительных слотов .....	49
1.2.1.3 Настройка форм виджетов созданных слотов.....	50
1.2.1.4 Создание экземпляров классов .....	53
1.2.1.5 Установка слотов отображения .....	54
1.2.1.6 Создание запроса .....	57
1.2.2. Задания для самостоятельной работы магистрантов.....	61
1.2.3. Контрольные вопросы .....	62
ЛАБОРАТОРНАЯ РАБОТА №2 .....	63
Введение.....	63
2.1. Часть 1. Выбор удельной нагрузки на крыло .....	65
2.1.1. Расчет условия обеспечения заданной скорости захода на посадку .....	65
2.1.2. Расчет условия обеспечения заданной крейсерской скорости на расчетной высоте полета .....	66
2.1.3. Расчет удельной нагрузки на крыло.....	67
2.1.4. Проектирование дескриптивной онтологии.....	68
2.1.4.1 Создание онтологии.....	68
2.1.4.2 Создание и удаление концепта .....	68
2.1.4.3 Создание связей между концептами .....	69

2.1.4.4	Представление онтологии в виде семантической сети .....	74
2.1.5.	Проектирование онтологии мира заказов и ресурсов .....	75
2.1.5.1	Создание онтологии мира заказов и ресурсов .....	75
2.1.5.2	Создание концепта «Агент заказа» .....	77
2.1.5.3	Создание концепта «Агент ресурса».....	77
2.1.5.4	Виртуальные отношения: отношения матчинга .....	78
2.1.5.5	Условия матчинга .....	79
2.1.5.6	Условия принятия (decision making machine conditions).....	80
2.1.5.7	Параметры представления экземпляра объекта в сцене .....	81
2.1.5.8	Сохранение онтологий предметной области “выбор удельной нагрузки на крыло” .....	82
2.1.6.	Создание онтологической сцены.....	83
2.1.7.	Моделирование сцены виртуального мира .....	85
2.1.7.1.	Запуск сцены на моделирование для самолета ТУ-154 .....	85
2.1.7.2.	Сохранение сцены виртуального мира .....	87
2.1.7.3.	Загрузка ранее созданной сцены виртуального мира.....	87
2.1.7.4.	Моделирование сцены для самолета ИЛ-86.....	87
2.1.7.5.	Моделирование сцены для самолета ЯК-40 .....	88
2.2.	Часть 2. Выбор тяговооруженности проектируемого самолета .....	89
2.2.1.	Выбор значений потребной тяговооруженности самолета.....	89
2.2.1.1.	Расчет тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости полета на расчетной высоте .....	89
2.2.1.2.	Расчет тяговооруженности из условия заданной длины разбега .....	90
2.2.1.3.	Расчет тяговооруженности из условия взлета при отказе одного двигателя.....	90
2.2.1.4.	Расчет потребной тяговооруженности самолета .....	90
2.2.2.	Проектирование дескриптивной онтологии.....	91
2.2.2.1.	Создание отологии.....	91
2.2.2.2.	Создание отологии.....	91
2.2.2.3.	Создание связей между концептами .....	92
2.2.2.4.	Представление онтологии в виде семантической сети .....	97
2.2.3.	Проектирование онтологии мира заказов и ресурсов .....	98
2.2.3.1.	Создание онтологии мира заказов и ресурсов .....	98
2.2.3.2.	Создание концепта «Агент заказа» .....	100
2.2.3.3.	Создание концепта «Агент ресурса».....	100
2.2.3.4.	Виртуальные отношения: отношение матчинга .....	101
2.2.3.5.	Условия матчинга .....	102
2.2.3.6.	Условия принятия решения (decision making machine conditions) .....	103
2.2.3.7.	Создание условия принятия решения - выбор максимального значения тяговооруженности .....	103
2.2.3.8.	Параметры представления экземпляра объекта в сцене .....	104
2.2.3.9.	Сохранение онтологий предметной области “Выбор потребной тяговооруженности самолета” .....	104
2.2.4.	Создание онтологической сцены.....	105
2.2.5.	Моделирование сцены виртуального мира .....	107
2.2.5.1.	Запуск сцены на моделирование для самолета ТУ-154 .....	107
2.2.5.2.	Сохранение сцены виртуального мира .....	108
2.2.5.3.	Загрузка ранее созданной сцены виртуального мира.....	109
2.2.5.4.	Моделирование сцены для самолета ИЛ-86.....	109
2.2.5.5.	Моделирование сцены для самолета ЯК-40.....	110
2.2.6.	Контрольные вопросы .....	111

ЛАБОРАТОРНАЯ РАБОТА №1 .....	10
1.1. Часть 1. Выбор транспортного средства .....	10
3.3. ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ №1 .....	11
3.4. СОЗДАНИЕ ОНТОЛОГИИ .....	11
3.2.1. Создание проекта .....	11
3.2.2. Сохранение проекта .....	12
3.2.3. Создание классов .....	13
3.2.4. Создание класса «Авиатранспорт» .....	14
3.2.5. Создание классов «Ж/Д транспорт», «Автомобильный транспорт» и «Речной транспорт» .....	14
3.2.6. Редактирование дерева иерархии .....	15
3.2.7. Создание слотов .....	16
3.2.8. Включение слотов в класс .....	18
3.2.9. Связывание слота с классом .....	20
3.2.10. Создание отношения между классами .....	22
3.2.11. Настройка формы ввода экземпляров .....	23
3.2.12. Изменение параметров виджета .....	24
Изменение размера виджета .....	24
Перемещение виджета .....	25
Изменение кнопок виджета .....	26
Изменение видимости виджета .....	27
Отображение скрытого виджета .....	28
3.2.13. Создание экземпляров класса .....	30
3.2.14. Установка слота отображения .....	32
3.2.15. Изменение параметров виджета для создания семантического представления онтологии .....	32
3.2.16. Создание семантического представления онтологии .....	37
3.2.17. Создание запроса .....	41
3.2.18. Сохранение запроса .....	46
3.2.19. Загрузка запроса .....	47
3.5. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ .....	47
1.2. Часть 2. Выбор самолета .....	49
3.1. Расширение онтологии .....	49
1.2.1.1 Загрузка проекта .....	49
1.2.1.2 Создание дополнительных слотов .....	49
1.2.1.3 Настройка форм виджетов созданных слотов .....	50
1.2.1.4 Создание экземпляров классов .....	53
1.2.1.5 Установка слотов отображения .....	54

1.2.1.6	Создание запроса .....	57
3.2.	ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ .....	61
3.3.	КОНТРОЛЬНЫЕ ВОПРОСЫ.....	62
	ЛАБОРАТОРНАЯ РАБОТА №2 .....	63
	ВВЕДЕНИЕ .....	63
1.1.	Часть 1. Выбор удельной нагрузки на крыло .....	65
2.1.1.	Расчет условия обеспечения заданной скорости захода на посадку .....	65
2.1.2.	Расчет условия обеспечения заданной крейсерской скорости на расчетной высоте полета .....	66
2.1.3.	Расчет удельной нагрузки на крыло.....	67
2.1.4.	Проектирование дескриптивной онтологии.....	68
2.1.4.1	Создание онтологии.....	68
2.1.4.2	Создание и удаление концепта .....	68
2.1.4.3	Создание связей между концептами .....	69
2.1.4.4	Представление онтологии в виде семантической сети .....	74
2.1.5.	Проектирование онтологии мира заказов и ресурсов .....	75
2.1.5.1	Создание онтологии мира заказов и ресурсов .....	75
2.1.5.2	Создание концепта «агент заказа» .....	77
2.1.5.3	Создание концепта «агент ресурса».....	77
2.1.5.4	Виртуальные отношения: отношения матчинга .....	78
2.1.5.5	Условия матчинга .....	79
2.1.5.6	Условия принятия (Decision Making Machine conditions) .....	80
2.1.5.7	Параметры представления экземпляра объекта в сцене .....	81
2.1.5.8	Сохранение онтологий предметной области “Выбор удельной нагрузки на крыло” .....	82
2.1.6.	Создание онтологической сцены.....	83
2.1.7.	Моделирование сцены виртуального мира .....	85
2.1.7.1.	Запуск сцены на моделирование для самолета Ту-154 .....	85
2.1.7.2.	Сохранение сцены виртуального мира .....	87
2.1.7.3.	Загрузка ранее созданной сцены виртуального мира.....	87
2.1.7.4.	Моделирование сцены для самолета Ил-86 .....	87
2.1.7.5.	Моделирование сцены для самолета Як-40.....	88
1.2.	Часть 2. Выбор тяговооруженности проектируемого самолета .....	89
2.2.1.	Выбор значений потребной тяговооруженности самолета.....	89
2.2.1.1.	Расчет тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости полета на расчетной высоте .....	89
2.2.1.2.	Расчет тяговооруженности из условия заданной длины разбега .....	90
2.2.1.3.	Расчет тяговооруженности из условия взлета при отказе одного двигателя.....	90
2.2.1.4.	Расчет потребной тяговооруженности самолета .....	90
2.2.2.	Проектирование дескриптивной онтологии.....	91
2.2.2.1.	Создание отологии.....	91
2.2.2.2.	Создание отологии.....	91
–	Концепт «объект» .....	91
–	Концепт «атрибут» .....	92
2.2.2.3.	Создание связей между концептами .....	92
2.2.2.4.	Представление онтологии в виде семантической сети .....	97
2.2.3.	Проектирование онтологии мира заказов и ресурсов .....	98
2.2.3.1.	Создание онтологии мира заказов и ресурсов .....	98
2.2.3.2.	Создание концепта «агент заказа» .....	100
2.2.3.3.	Создание концепта «агент ресурса».....	100
2.2.3.4.	Виртуальные отношения: отношение матчинга .....	101
2.2.3.5.	Условия матчинга .....	102

2.2.3.6.	Условия принятия решения (Decision Making Machine conditions) .....	103
2.2.3.7.	Создание условия принятия решения - выбор максимального значения тяговооруженности .....	103
2.2.3.8.	Параметры представления экземпляра объекта в сцене .....	104
2.2.3.9.	Сохранение онтологий предметной области “Выбор потребной тяговооруженности самолета” .....	104
2.2.4.	Создание онтологической сцены.....	105
2.2.5.	Моделирование сцены виртуального мира .....	107
2.2.5.1.	Запуск сцены на моделирование для самолета Ту-154 .....	107
2.2.5.2.	Сохранение сцены виртуального мира .....	108
2.2.5.3.	Загрузка ранее созданной сцены виртуального мира.....	109
2.2.5.4.	Моделирование сцены для самолета Ил-86 .....	109
2.2.5.5.	Моделирование сцены для самолета Як-40.....	110
2.2.6.	Контрольные вопросы .....	111
3.	ЛАБОРАТОРНАЯ РАБОТА №3 .....	112
	ВВЕДЕНИЕ .....	112
3.1.	ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ №3 .....	112
3.2.	СОЗДАНИЕ ОНТОЛОГИИ .....	113
3.2.1.	<i>Создание проекта .....</i>	<i>113</i>
3.2.2.	<i>Сохранение проекта.....</i>	<i>114</i>
3.2.3.	<i>Создание классов .....</i>	<i>116</i>
3.2.4.	<i>Создание класса «Самолет».....</i>	<i>116</i>
3.2.5.	<i>Создание подклассов класса «Самолет».....</i>	<i>117</i>
3.2.6.	<i>Создание суперклассов .....</i>	<i>119</i>
3.2.7.	<i>Создание свойств объекта и данных об объекте .....</i>	<i>122</i>
3.2.8.	<i>Создание экземпляров .....</i>	<i>123</i>
3.2.9.	<i>Задание значений для экземпляров .....</i>	<i>125</i>
3.2.10.	<i>Создание семантического представления онтологии .....</i>	<i>127</i>
3.2.11.	<i>Сохранение семантической сети.....</i>	<i>128</i>
3.3.	ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ.....	130
	ЛИТЕРАТУРА .....	13030

## ПРЕДИСЛОВИЕ

Курс лабораторных работ по дисциплине «Онтология проектирования» состоит из двух частей. В первой части решаются типовые проектные задачи выбора и принятия решений (Лабораторные работы № 1 и № 2), во второй – осуществляется онтологический анализ предметной области на примере самолета (Лабораторная работа № 3). В качестве конструкторов и редакторов онтологий используются программные продукты Magenta (Magenta Corporation Limited) и Protégé (Стэнфордский университет США).

Выполнение первой и третьей лабораторных работ осуществляется с помощью открытого онтологического редактора **Protégé**. Платформа Protégé поддерживает два основных способа моделирования онтологий посредством редакторов Protégé-Frames (Protégé 3.3 – Лабораторная работа №1) и Protégé-OWL (Protégé 4.1.beta – Лабораторная работа №3). Онтологии, построенные в Protégé, могут быть экспортированы во множество форматов, включая RDF, OWL и XML (<http://protege.stanford.edu/>).

Проектирование технической системы – сложный наукоемкий процесс, которому предшествует этап анализа потребности. Потребностью, которую реализуют рассматриваемые в курсе лабораторных работ транспортные системы, является перевозка грузов. Обеспечение условий качественной перевозки грузов является одной из главных задач проектирования.

Важным аспектом при проектировании является онтологический анализ предметной области, суть которого заключается в структуризации и классификации сущностей, установлении связей, параметров и характеристик и разработки тезауруса. Онтологический анализ, помимо упорядочивания знаний о предметной области, также способствует повышению качества проектных работ.

В **первой** лабораторной работе решается задача перевозки грузов из одного населенного пункта в другой. В качестве начального и конечного пунктов маршрута следования грузов принимаются города Москва и Самара соответственно. Существует множество способов решения этой задачи, различных по следующим критериям:

- грузоподъемность транспортного средства;
- объем перевозимого груза;
- стоимость и время перевозки и другие.

В соответствии с начальными условиями и заданными параметрами находится наиболее эффективный способ перевозки груза с помощью одного из рассматриваемых транспортных средств:

- авиатранспорт;
- автотранспорт;
- железнодорожный транспорт;
- речной транспорт.

Онтологический анализ сущностей, характеристик и связей между ними позволяет найти оптимальный способ доставки полезного груза.

После выбора эффективного вида транспортного средства осуществляется дальнейшая конкретизация и детализация. Онтология дополняется реально существующими марками самолетов. В этом случае содержание технического задания становится более детальным, с привязкой к особенностям выбранного транспортного средства.

Во **второй** лабораторной работе на основании заданных характеристик самолетов-прототипов, информация о которых хранится в базе данных, осуществляется выбор удельной нагрузки на крыло и потребной тяговооруженности самолета.

В качестве инструмента во второй лабораторной работе используется конструктор онтологий и исполнительная система **Magenta** (<http://www.magenta-technology.com/>).

Важной практической работой, выполняемой магистрантами в рамках дисциплины «Онтология проектирования», является **третья** лабораторная работа. В методических указаниях приведен пример выполнения онтологического анализа предметной области «самолет» в редакторе **Protégé 4.1 beta**. Эта версия Protégé позволяет построить онтологию для семантической сети на OWL (Web Ontology Language). Формальная семантика OWL дает возможность получения логических следствий, то есть фактов, которые не присутствуют непосредственно в онтологии, но могут быть выведены из существующих посредством семантики.

**Цель курса лабораторных работ** заключается в освоении метода онтологического анализа на примерах решения проектных задач по выбору транспортных систем и их характеристик и структуризации предметной области самолет.

**Задачи** лабораторного практикума:

- ознакомиться с современными редакторами и конструкторами онтологий **Protégé** (версии 3.3 и 4.1 beta) и **Magenta**;
- получить навыки решения проектных задач с помощью онтологических систем.

Лабораторные работы выполняются одновременно с усвоением материала учебных пособий [1, 2] и прочтением подборки научных статей [3], подготовленных для данной дисциплины.

## ЛАБОРАТОРНАЯ РАБОТА №1

### 1.1. Часть 1. Выбор транспортного средства

#### ВВЕДЕНИЕ

В литературе по искусственному интеллекту содержится много различных определений понятия «онтология» (в том числе [1-3]). Обобщенно, онтология, как способ представления знаний о фрагменте окружающего мира, есть точная спецификация некоторой предметной области. Онтология включает в себя словарь для представления и обмена знаниями о предметной области и множество связей, установленных между терминами в этом словаре. Использование онтологий как одного из методов систематизации терминов и понятий является средством описания знаний, объединяющим в себе другие известные модели представления знаний.

Онтология вместе с набором индивидуальных экземпляров классов образует базу знаний. Онтология всегда отражает взгляд аналитика, т.е. всегда субъективна, а поэтому не существует единственно правильного способа ее создания. Несмотря на это можно выделить некоторые основные принципы, на которых базируется создание онтологии:

*Ясность* — онтология должна эффективно передавать смысл введенных терминов, ее определения должны быть объективны, а для их объективизации должен использоваться четко фиксированный формализм.

*Согласованность* — все определения должны быть логически непротиворечивы.

*Расширяемость* — необходимо проектировать онтологию так, чтобы ее словари терминов можно было расширять, используя уже существующие понятия.

*Минимум влияния кодирования* — концептуализация онтологии должна быть специфицирована на уровне представления, а не символического кодирования.

*Минимум онтологических обязательств* — онтология должна содержать только наиболее существенные предположения о моделируемой предметной области, чтобы оставлять свободу расширения и специализации.

Protégé – один из наиболее популярных редакторов онтологий, созданный в Стэнфордском университете (США). В Protégé все понятия предметной области делятся на классы, подклассы, экземпляры.

Практически процесс построения онтологии в Protege включает в себя:

1. Определение классов в онтологии (classes),
2. Организация классов в некоторую иерархию,
3. Определение слотов (slots) и их допустимых значений,
4. Заполнение значений слотов для экземпляров (instances) классов.

### 3.3. ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ №1

#### Цели лабораторной работы:

1. Освоить работу с редактором онтологий Protege.
2. Освоить навыки создания онтологии.

#### Задачи работы:

1. Создать онтологию транспортных средств.
2. Наполнить созданную онтологию экземплярами.
3. Осуществить выбор транспортного средства для перевозки груза, отвечающего заданным требованиям.

Среди требований (характеристик транспортных средств) рассматриваются: грузоподъемность, объем грузового отсека, время перевозки, стоимость перевозки.

### 3.4. СОЗДАНИЕ ОНТОЛОГИИ

#### 3.2.1. Создание проекта

Работа в системе Protege начинается с создания нового проекта или выбора существующего.

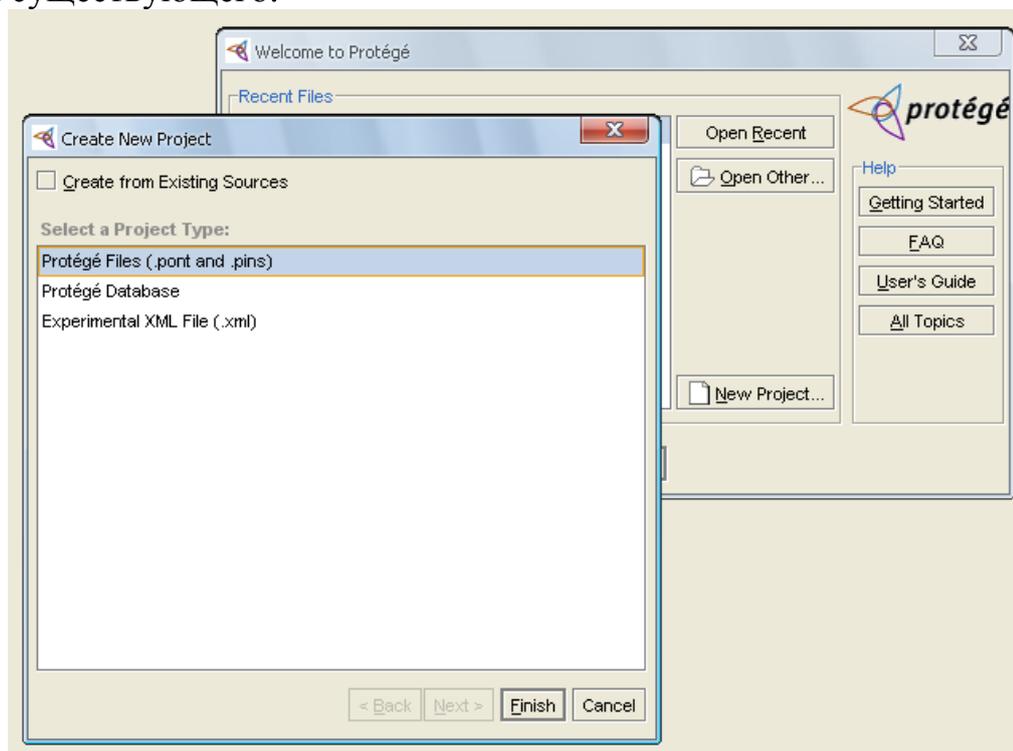


Рисунок 1.1 – Окно приветствия

Для создания нового проекта после запуска программы в окне приветствия (Рисунок 1.1) выберете кнопку «New Project...». Появится диалоговое окно "Create New Project", позволяющее выбрать тип проекта. Если нет необходимости создавать файлы в специальном формате, нажмите

кнопку Finish – будет выбран формат файла по умолчанию Protege Files (.pont and pins). Обратите внимание, что отметку в окне «Create From Existing Sources» ставить не нужно. После этого откроется окно проекта Protégé в области просмотра классов (Classes) (Рисунок 1.2). На данный момент в этой области находятся только внутренние системные классы Protégé THING и SYSTEM-CLASS.

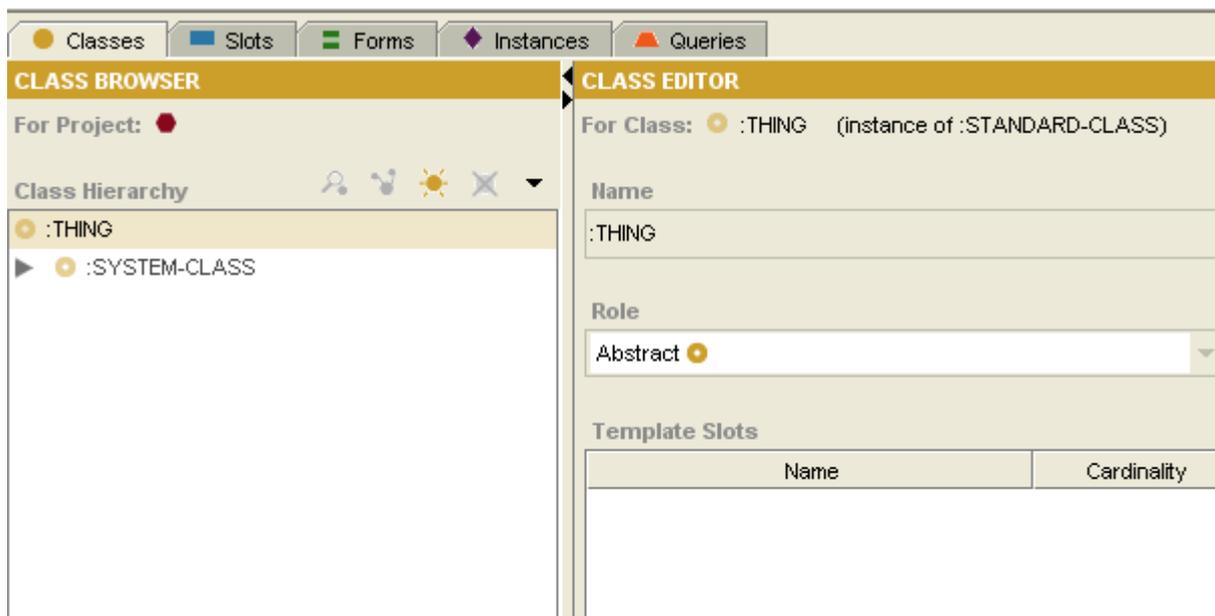


Рисунок 1.2 – Область просмотра классов

### 3.2.2. Сохранение проекта

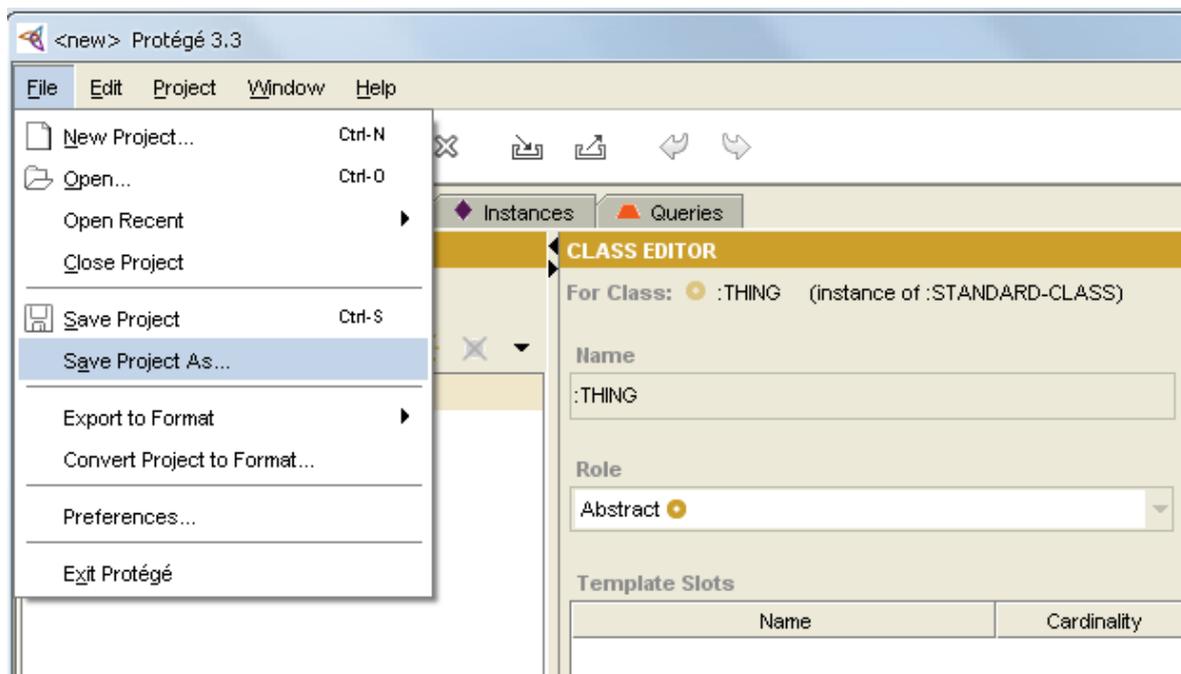


Рисунок 1.3 - Окно сохранения проекта

Сохранить проект можно несколькими способами:

1. выбрав из меню File пункт «Save project As...» (Рисунок 1.3),
2. нажав на панели инструментов кнопку сохранения проекта,
3. при последовательном нажатии клавиш Ctrl+S.

После этого появится окно выбора имени файла проекта (Рисунок 1.4).

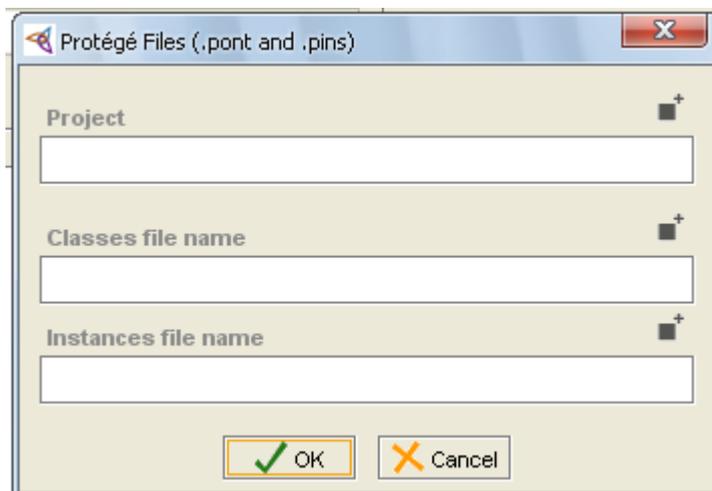


Рисунок 1.4 - Окно выбора имени файла проекта

Для того чтобы указать место, куда будет сохранен Ваш проект, нажмите кнопку , которая находится напротив надписи Project. В открывшемся диалоговом окне, перейдите по нужному пути в файловой системе или создайте каталог, где будут храниться данные проекта и откройте созданную папку. Введите имя файла проекта – «LRI\_varX», где X номер Вашего варианта. После нажатия кнопки Select вы вернетесь в окно сохранения файлов проекта. Нажмите ОК и проект будет сохранен.

### 3.2.3. Создание классов

Основное окно программы Protégé состоит из закладок (tabs) которые отображают различные аспекты модели знаний. При создании нового проекта наиболее важной закладкой является закладка классов (Classes). Классы соответствуют объектам или типам объектов, в некой предметной области. В лабораторной работе, классы будут включать в себя виды транспортных средств, а именно, авиатранспорт, железнодорожный, автомобильный и речной транспорт.

Классы в Protégé отображаются в виде иерархии наследования (Class Hierarchy), которая располагается в области просмотра называемой Class Browser (или навигатор классов) в левой части закладки классов. Свойства классов выбранных в текущий момент в навигаторе, будут отображены в редакторе классов справа в полях Name, Template Slots (Рисунок 1.2).

### 3.2.4. Создание класса «Авиатранспорт»

Для того чтобы создать новый класс в навигаторе классов (Class Browser), выделите класс THING (вещь, нечто). Все создаваемые классы должны находиться на уровень ниже этого системного класса. Класс SYSTEM\_CLASS используется для определения структур различных форм Protege.

Нажмите кнопку  создать класс (Create Class) в верхнем правом углу навигатора классов. Новый класс будет создан со стандартным именем, основанным на имени проекта. В поле Name при помощи клавиатуры введите название класса «Авиатранспорт» и нажмите клавишу Enter. Вы можете увидеть, что имя нового класса в навигаторе классов после создания будет выделено, для указания того, что этот класс выбран в данный момент (Рисунок 1.5).

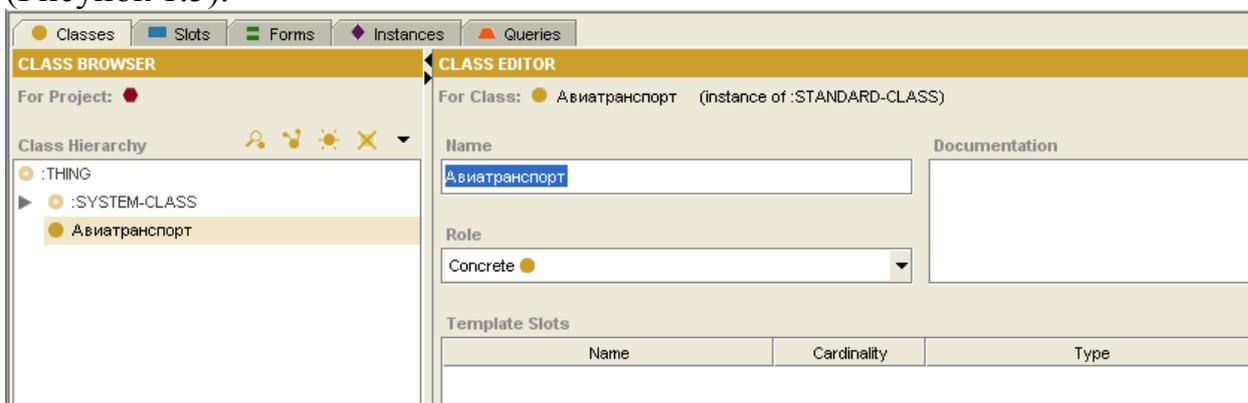


Рисунок 1.5 – Окно редактора класса

В системе Protege названия классов принято писать с заглавной буквы, а названия слотов - со строчной. Такое правило помогает отличить классы от слотов в созданной онтологии.

### 3.2.5. Создание классов «Ж/Д транспорт», «Автомобильный транспорт» и «Речной транспорт»

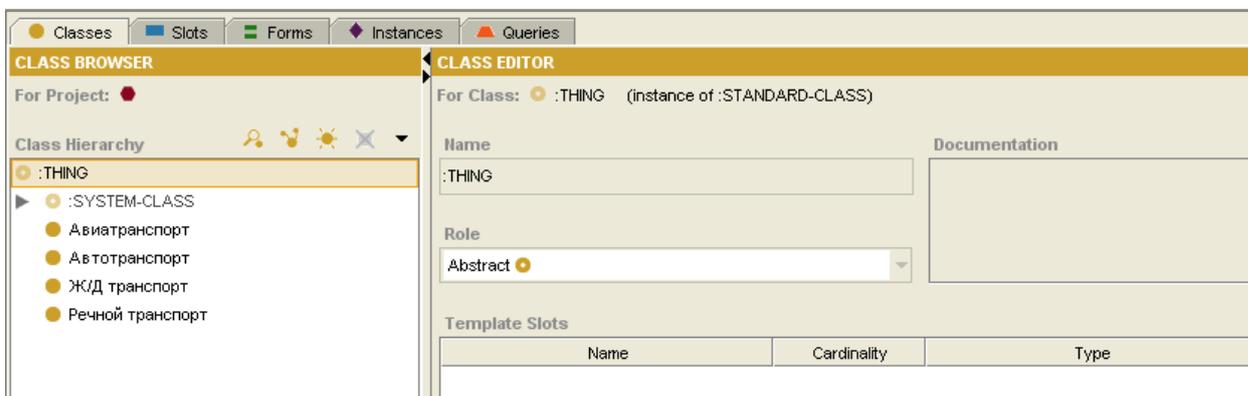


Рисунок 1.6 – Окно просмотра созданных классов

При создании новых классов в навигаторе Class Browser, необходимо выделять класс THING, для того, чтобы созданные классы находились на одном уровне. Аналогично предыдущему пункту создайте классы «Ж/Д транспорт», «Автомобильный транспорт» и «Речной транспорт» (Рис.1.6).

### 3.2.6. Редактирование дерева иерархии

В совокупности авиатранспорт, автомобильный транспорт, ж/д транспорт и речной транспорт представляют собой транспортную систему. Для того, чтобы отразить это в структуре онтологии создайте новый класс «Транспортная система». Этот класс будет находиться на одном уровне с ранее созданными классами (Рисунок 1.7).

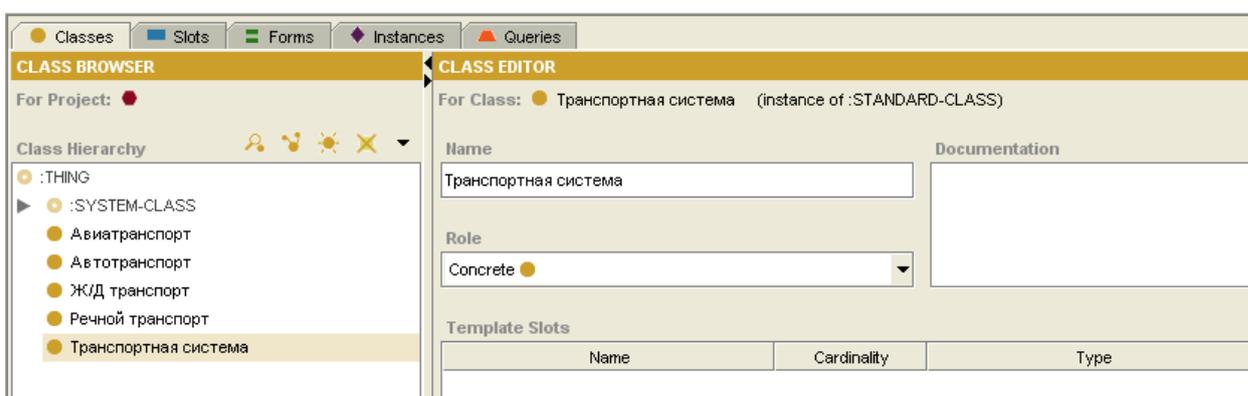


Рисунок 1.7 – Создание класса «Транспортная система»

На самом деле классы «Авиатранспорт», «Автомобильный транспорт», «Ж/Д транспорт», и «Речной транспорт» должны быть на уровень ниже класса «Транспортная система». Этот пример приведен для того, чтобы научиться редактировать структуру дерева иерархии.

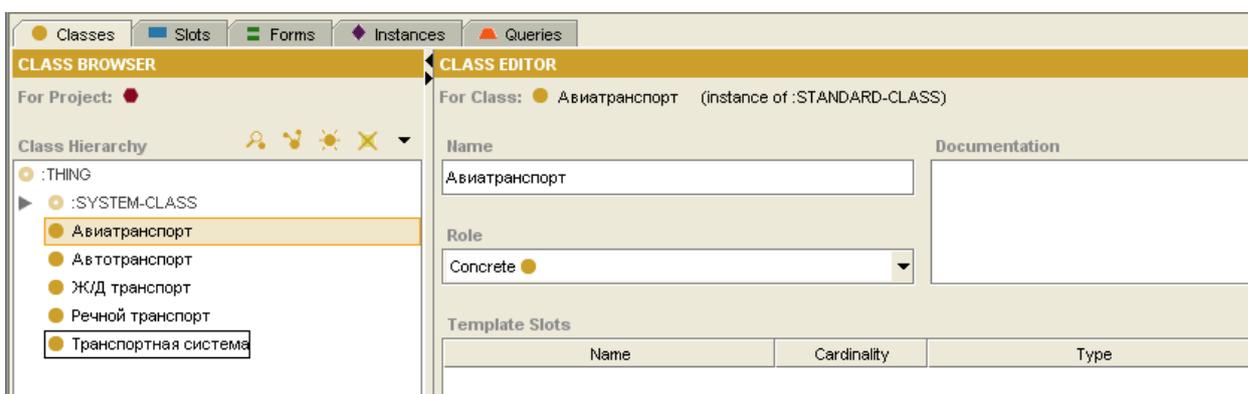


Рисунок 1.8 – Включение класса «Авиатранспорт» в класс «Транспортная система»

Для того чтобы изменить наследование в дереве иерархии, выберите класс «Авиатранспорт» и при нажатой левой клавиши мыши «перетащите» его в класс «Транспортная система» (Рисунок 1.8).

То же самое сделайте для других классов. Таким образом, класс «Транспортная система» будет содержать подклассы «Авиатранспорт», «Автомобильный транспорт», «Ж/Д транспорт», и «Речной транспорт» (Рисунок 1.9).

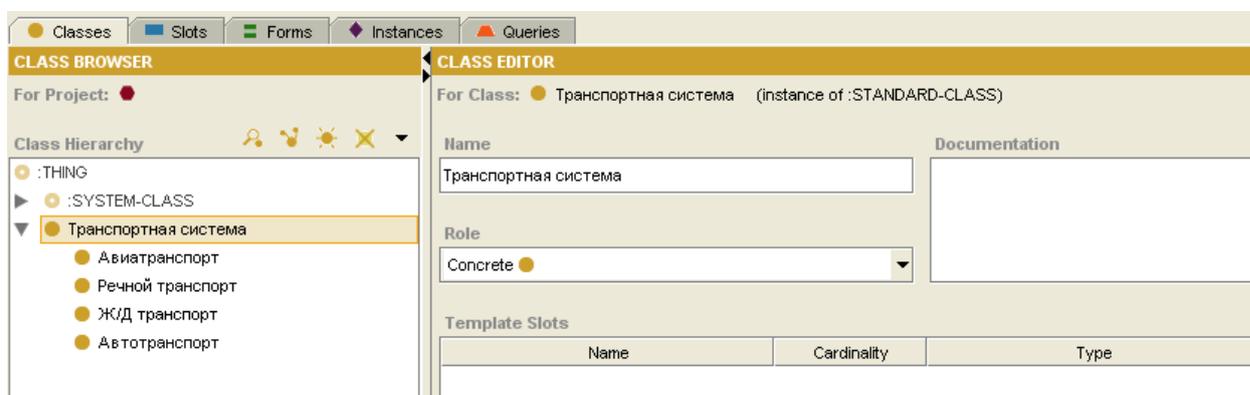


Рисунок 1.9 - Отредактированное дерево иерархии

Для дальнейшего графического представления, создайте класс «Семантическая сеть» на одном уровне с классом «Транспортная система» (Рисунок 1.10).

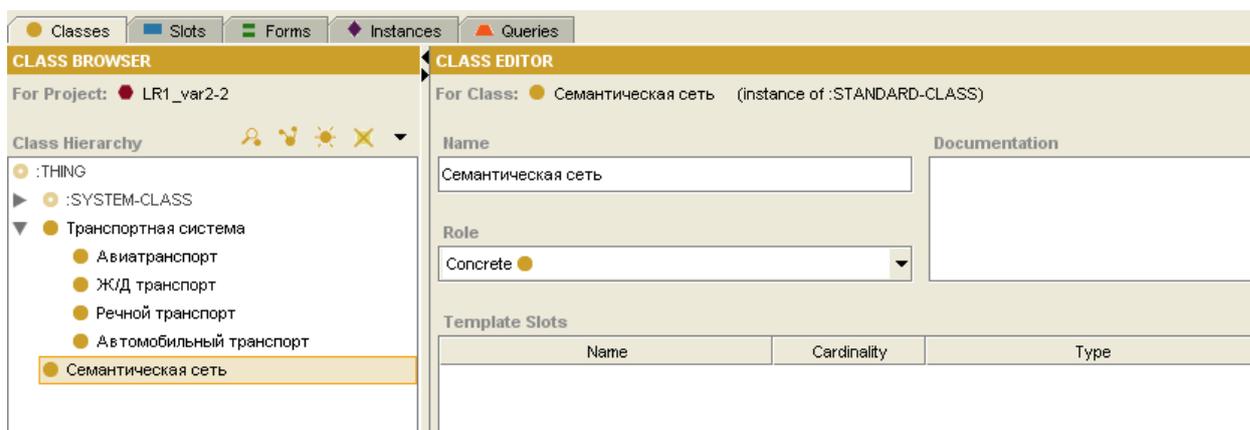


Рисунок 1.10 – Создание класса «Семантическая сеть»

### 3.2.7. Создание слотов

В системе Protege под классами понимаются конкретные понятия (концепции) предметной области, такие как Авиатранспорт или Автомобильный транспорт. В то же время классы это больше чем объекты, объединенные в иерархию. Они также могут иметь атрибуты (свойства) и отношения между ними, например, грузоподъемность, стоимость, время перевозки, объем грузового отсека транспортного средства и др.

Атрибуты и отношения класса описываются конструкцией под названием слот. В данном разделе будет показано, как создавать слоты, привязывать слоты к классам, описывать отношения между классами, а также будет описан механизм наследования слотов.

Управление на закладке слотов, схоже с закладкой классов, созданные слоты отображаются слева в области просмотра, а их редактирование возможно при помощи редактора справа.

Для создания слота существует несколько способов, рассмотрим один из них.

В основном окне программы необходимо выбрать закладку Slots, после чего нажать на кнопку  Create Slot в правом верхнем углу панели отображения иерархии слотов Slot Hierarchy (Рисунок 1.11).

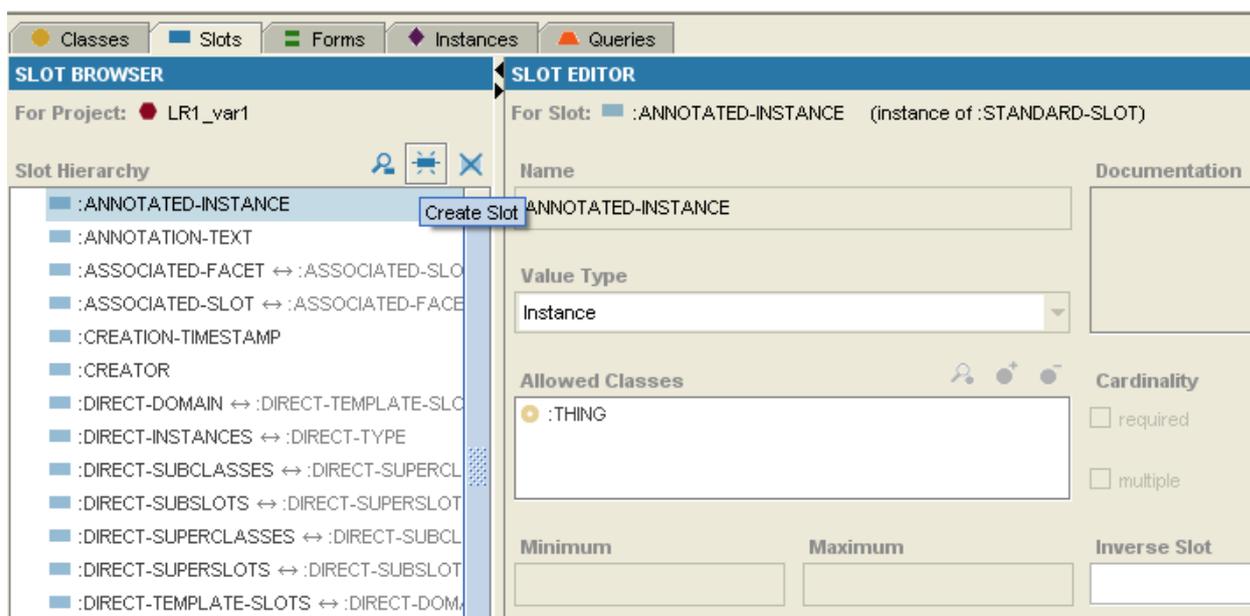


Рисунок 1.11 – Окно создания слота

Будет создан новый слот. Также как и при создании класса, ему будет присвоено стандартное имя. Переименуйте его в «название». Слот имеет по умолчанию тип значения (Value Type) - String (строка). Тип накладывает ограничения на то, какие значения может принимать слот. Строковый слот, к примеру, может принимать в качестве значений алфавитно-цифровые строки (включая пробелы). Для этого простого слота нужно лишь поставить отметку в графе Multiple, для того, чтобы его можно было связать с несколькими классами, менять остальные аспекты (facets) в редакторе слотов не нужно (Рисунок 1.12).

Аналогичным образом создайте слоты «грузоподъемность(кг)», «объем отсека(м3)», «время(ч)», «стоимость(руб/кг)». Эти слоты не будут являться строковыми, поэтому в графе Value Type необходимо выбрать тип Float и в окне Minimum поставить значение «0» для того, чтобы быть уверенными, что любое значение для этих полей не будет отрицательным.

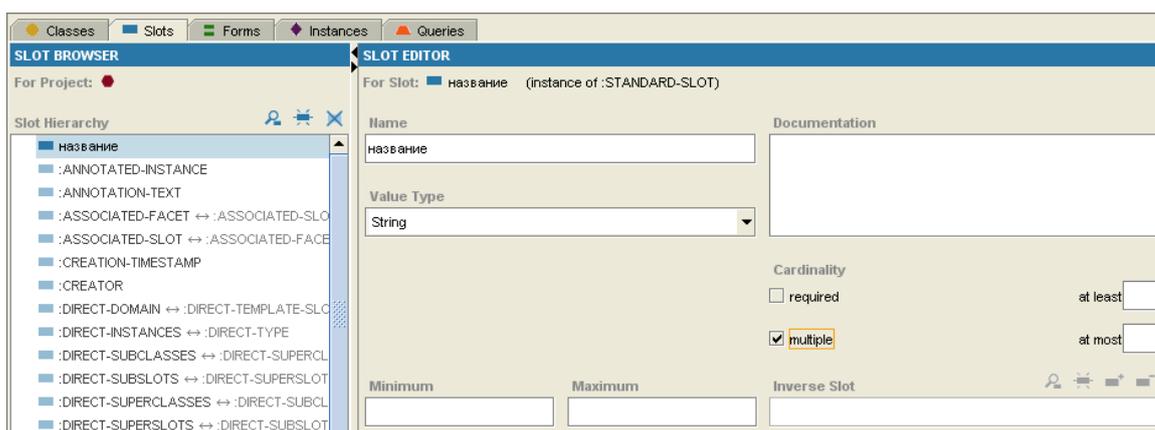


Рисунок 1.12 – Создание слота «название»

Также создайте слот «круглогодичность» в качестве типа выберите «String».

Для всех созданных слотов поставьте отметку в графе Multiple (Рисунок 1.13).

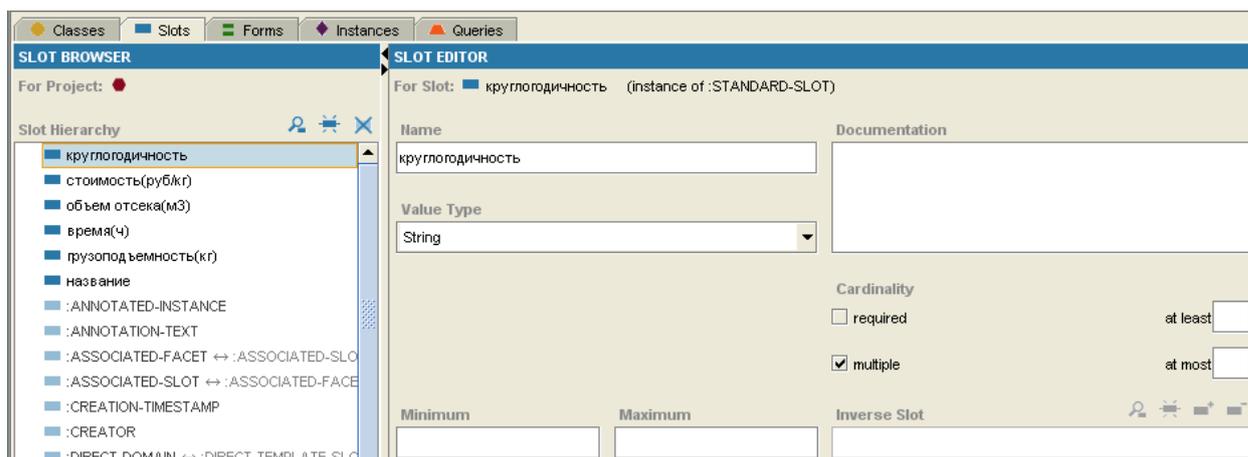


Рисунок 1.13 – Добавление слотов

### 3.2.8. Включение слотов в класс

На этом этапе Вы определили общий атрибут «название». Для того чтобы задействовать его в создаваемой онтологии, необходимо привязать его к классу.

Вернитесь к закладке классов (Classes) и откройте на редактирование класс «Транспортная система». Любой созданный или связанный с классом атрибут отображается в редакторе классов, справа от навигатора классов, пока в этой области нет записей. Выберите класс «Транспортная система» для того чтобы связать с ним слот «название». В редакторе классов на панели шаблонов слотов (Template slots), нажмите кнопку  Add Slot, после этого появится диалог выбора слота, в котором будет отображен список всех доступных слотов в Вашем проекте в алфавитном порядке, за исключением системных классов Protege, которые будут видны в самом низу списка. Выберите слот «название» (Рисунок 1.14).

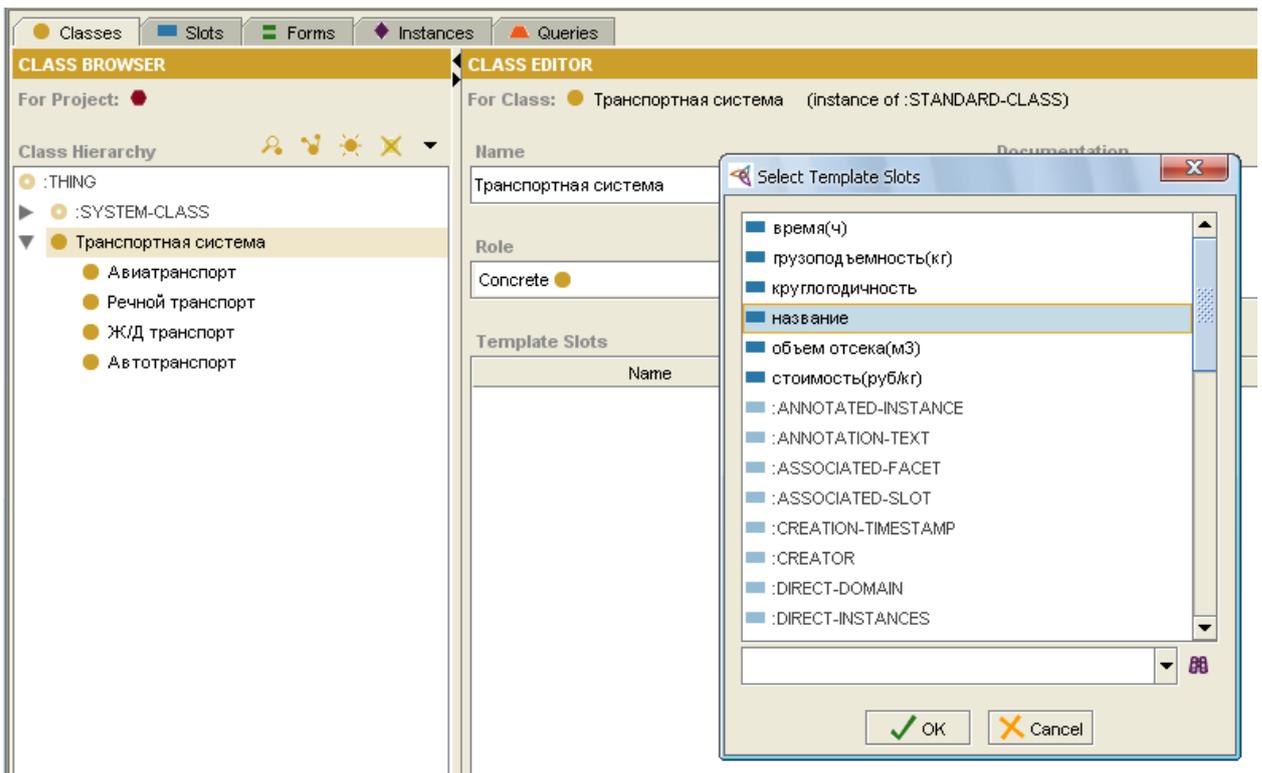


Рисунок 1.14 – Включение слота в класс

Для включения нескольких слотов сразу в окне выбора (Select Template Slot), выделите необходимые слоты при нажатой клавише Ctrl и нажмите ОК.

На панели шаблонов слотов (Template slots) теперь можно видеть все слоты, включенные в класс «Транспортная система», а также их свойства: мощность (количество элементов типа) и сам тип (String, Float) и минимальное значение (Minimum) (Рисунок 1.15).

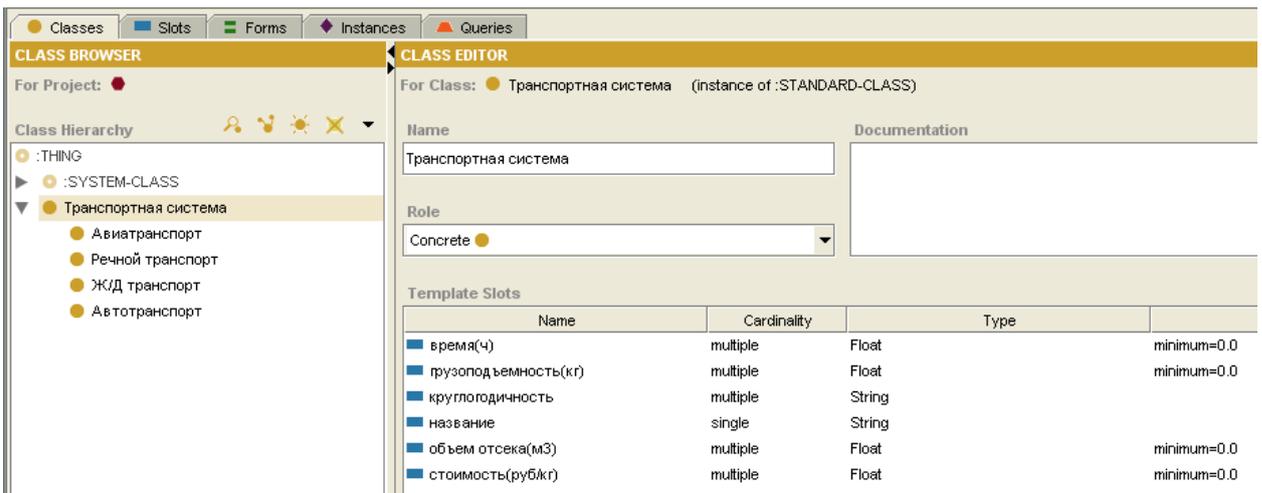


Рисунок 1.15 – Слоты класса «Транспортная система»

Поскольку «Авиатранспорт», «Автомобильный транспорт», «Ж/Д транспорт», и «Речной транспорт» являются подклассами класса

«Транспортная система» и имеют те же слоты, они отображаются в окне Template Slots в скобках (Рисунок 1.16).

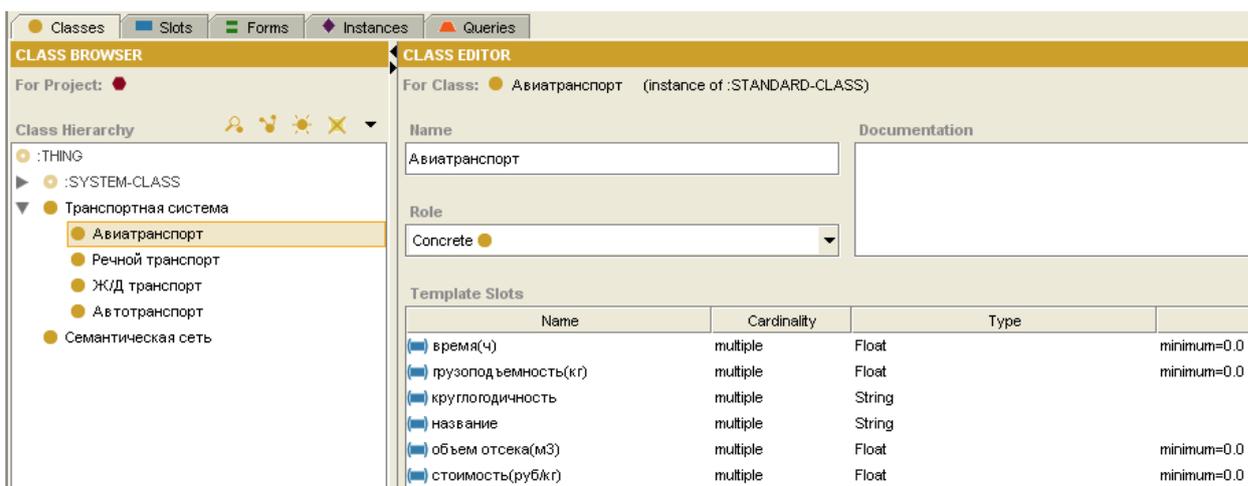


Рисунок 1.16 – Слоты класса «Авиатранспорт»

### 3.2.9. Связывание слота с классом

В пункте 1.8. говорилось о том, как для класса выбрать необходимые слоты. Связывать слот с классом можно и другим способом. Откройте вкладку «Slots» и выберете слот «название». На данном этапе в свойствах слота «Slot Editor» в окне «Domain» находится только класс «Транспортная система» (Рисунок 1.17).

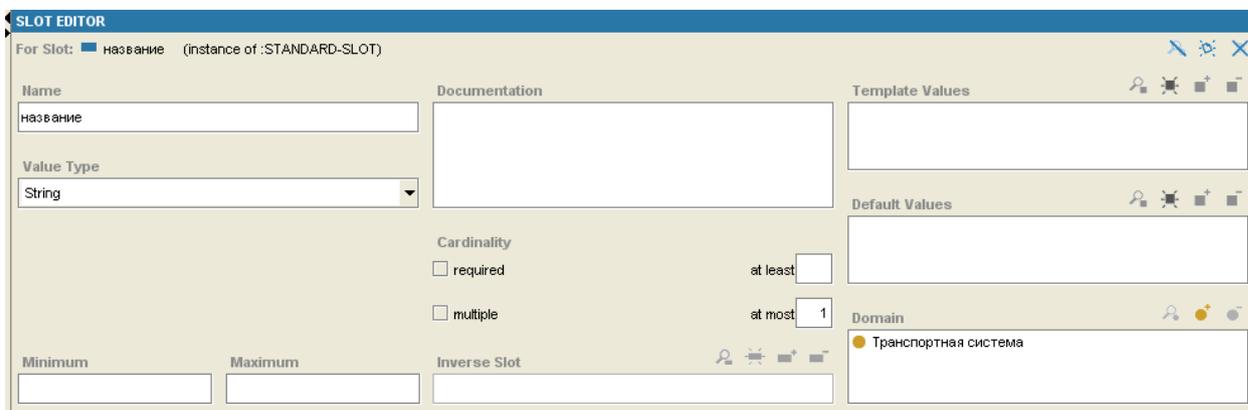


Рисунок 1.17 – Окно свойств слота «Название»

Для добавления класса нажмите кнопку  Add Class и из предложенного списка выберете класс «Семантическая сеть» и нажмите ОК (Рисунок 1.18).

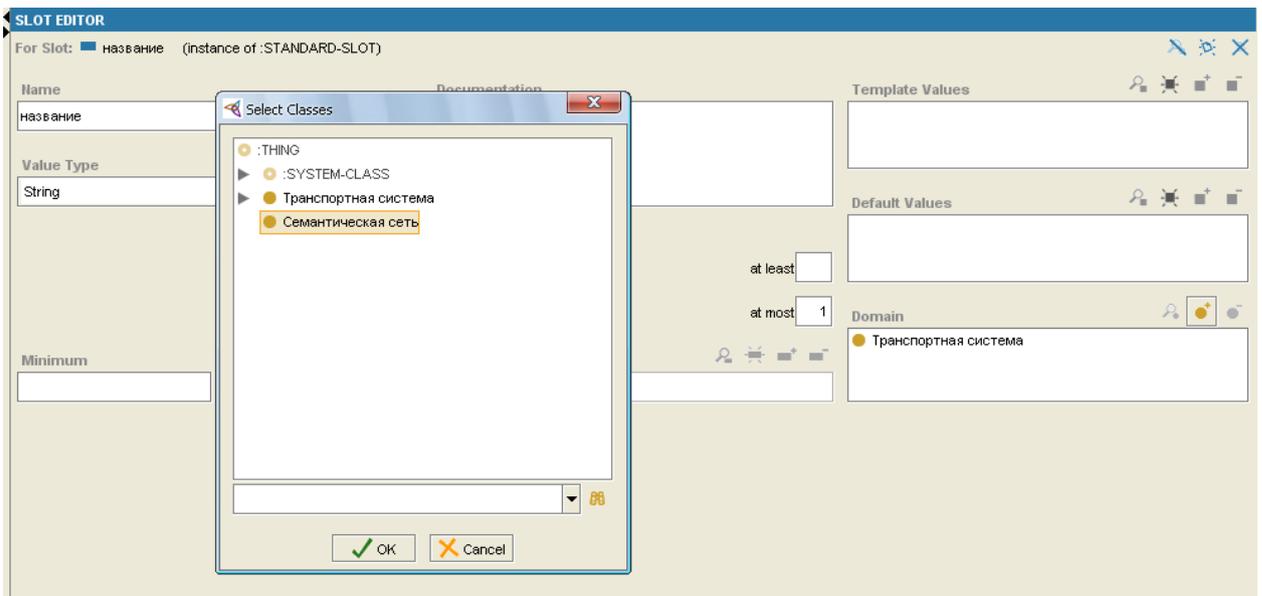


Рисунок 1.18 – Связывание слота с классом

Теперь в свойствах слота «название» в окне Domain находятся два класса «Транспортная система» и «Семантическая сеть» (Рисунок 1.19).

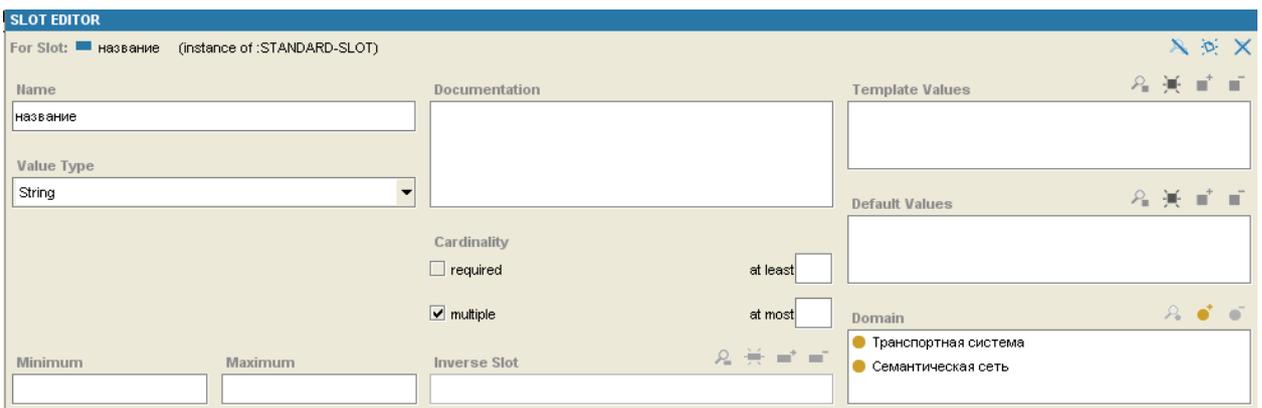


Рисунок 1.19 – Отредактированные свойства слота «название»

Если Вы перейдете во вкладку Classes, то увидите, что в свойствах класса «Семантическая сеть» теперь находится слот «название» (Рисунок 1.20).

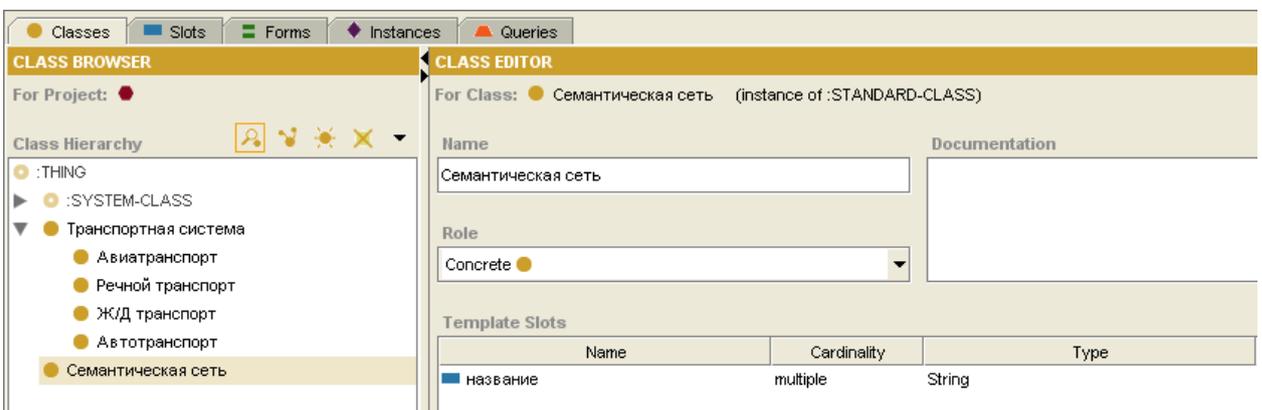


Рисунок 1.20 – Свойства класса «Семантическая сеть»

### 3.2.10. Создание отношения между классами

Система Protege позволяет создавать слоты, которые могут быть использованы для описания отношений между классами, которые не определены в иерархии классов. Для этого существуют слоты Instance (экземпляр) или Class (класс). Например, для графического представления онтологии был создан класс «Семантическая сеть», теперь необходимо его связать с классом «Транспортная система» таким образом, что бы это не противоречило создаваемой онтологии. Для этого можно создать новый слот, который будет описывать связь между классами.

В навигаторе классов выберете класс «Семантическая сеть». Нажмите кнопку  Create Slot для того чтобы создать и связать новый слот с этим классом. В открывшейся форме редактирования, наберите в поле имя (Name) «организационная структура», в строке тип значения (Value Type) выберете «Instance» (экземпляр) и в окне выбора доступных классов (Allowed Classes) при помощи кнопки  Add Class из предложенного списка выберете класс «Транспортная система». Чтобы в семантической сети было более одной связи, поставьте галочку в пункте multiple, на панели мощности (Cardinality) (Рисунок 1.21). После создания свойств слота закройте окно, свойства автоматически будут сохранены.

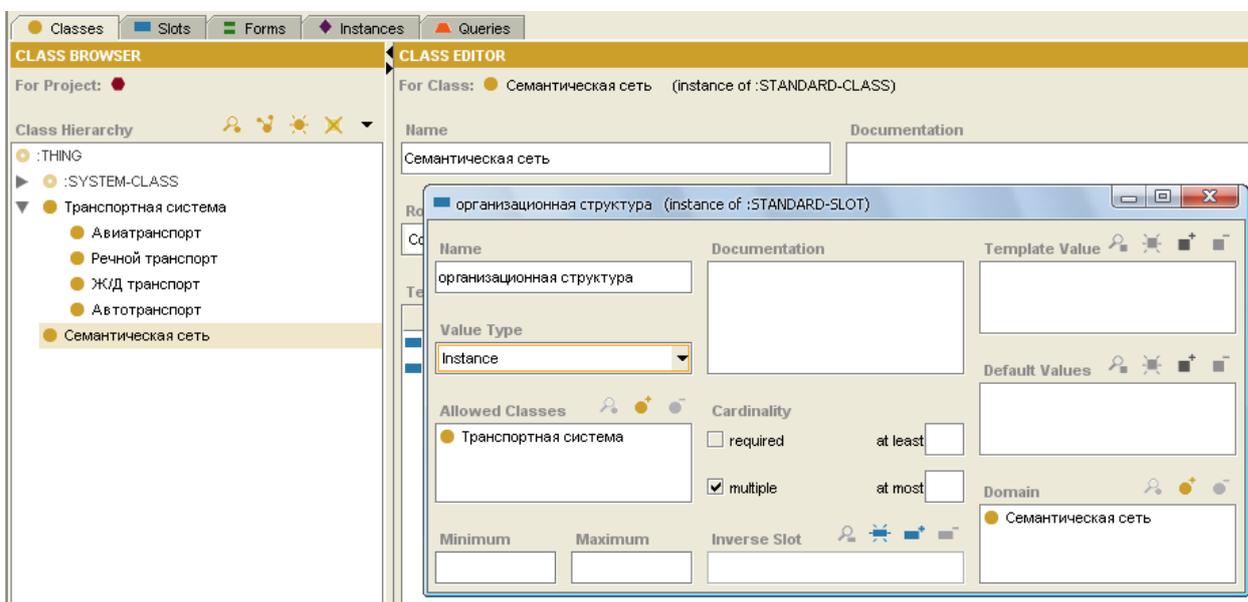


Рисунок 1.21 – Создание слота «организационная структура»

Для класса «Транспортная система» аналогичным образом создайте слот «есть» и установите для него такие же свойства, как и для предыдущего слота.

Теперь с помощью созданных слотов можно представить онтологию в виде семантической сети, в которой будут отображены экземпляры класса «Транспортная система» и связи между ними (п.1.1.2.15.)

### 3.2.11. *Настройка формы ввода экземпляров*

Для каждого класса онтологии, Protege генерирует форму по умолчанию, которую можно использовать для ввода данных экземпляра. Формы содержат поля ввода данных, называемые виджетами для каждого слота связанного с классом. Для разных типов данных слотов существуют разные типы виджетов, например:

- текстовый - TextFieldWidget, для слотов с типом данных строка,
- целочисленный - IntegerFieldWidget для полей, у которых значение представлено как целое число,
- список экземпляров – InstanceListWidget, для слотов, у которых в качестве типа установлен экземпляр класса и при этом мощность (количество элементов) больше одного и т.д.

В Protege существует возможность изменять стандартную форму представления экземпляров, если по какой либо причине она Вам не подходит. Можно изменить имя, размер, тип, местоположение и видимость виджета.

Откройте закладку форм (Forms) и выберете класс «Транспортная система». Вы увидите стандартную форму представления экземпляров (Рисунок 1.22).

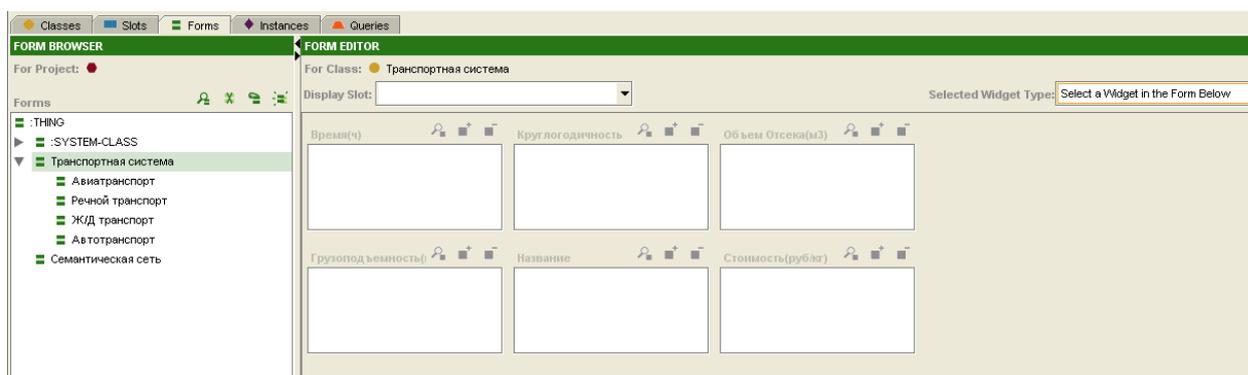


Рисунок 1.22 - Стандартная форма виджетов

Для того чтобы увидеть, как изменения, которые будут сделаны на закладке форм, отобразятся в редакторе экземпляров, перейдите на закладку экземпляров (Instances), выберете «Транспортная система», в окне Instances Browser нажмите кнопку  Create Instances. В окне Instances Editor, которое до этого момента было пустым, появятся виджеты в той же форме, что и в закладке Forms (Рисунок 1.23).

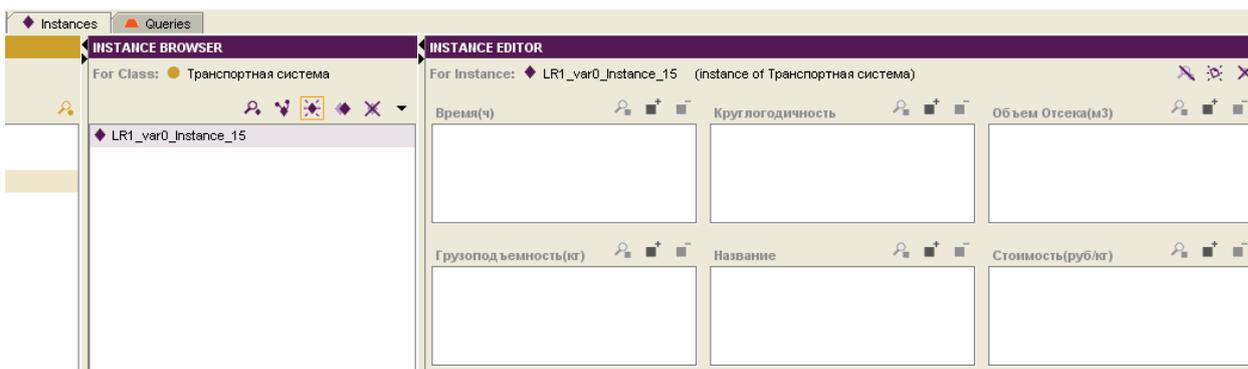


Рисунок 1.23 – Окно экземпляров

### 3.2.12. *Изменение параметров виджета*

#### **Изменение размера виджета**

Для того, чтобы изменить размер виджета, кликните на него мышью, чтобы вокруг появилась зеленая рамка (Рисунок 1.24). Это значит, что виджет выбран и, можно начать его редактирование.

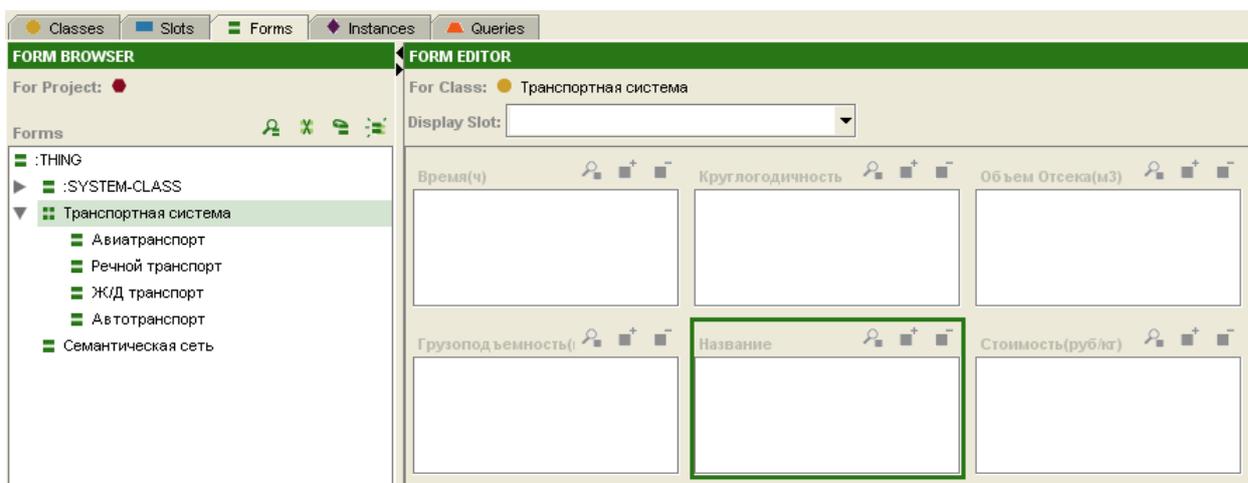


Рисунок 1.24 – Выбор виджета для редактирования

Щелкните по границе виджета и, удерживая кнопку мыши нажатой, перетащите ее, чтобы изменить размер (Рисунок 1.25).

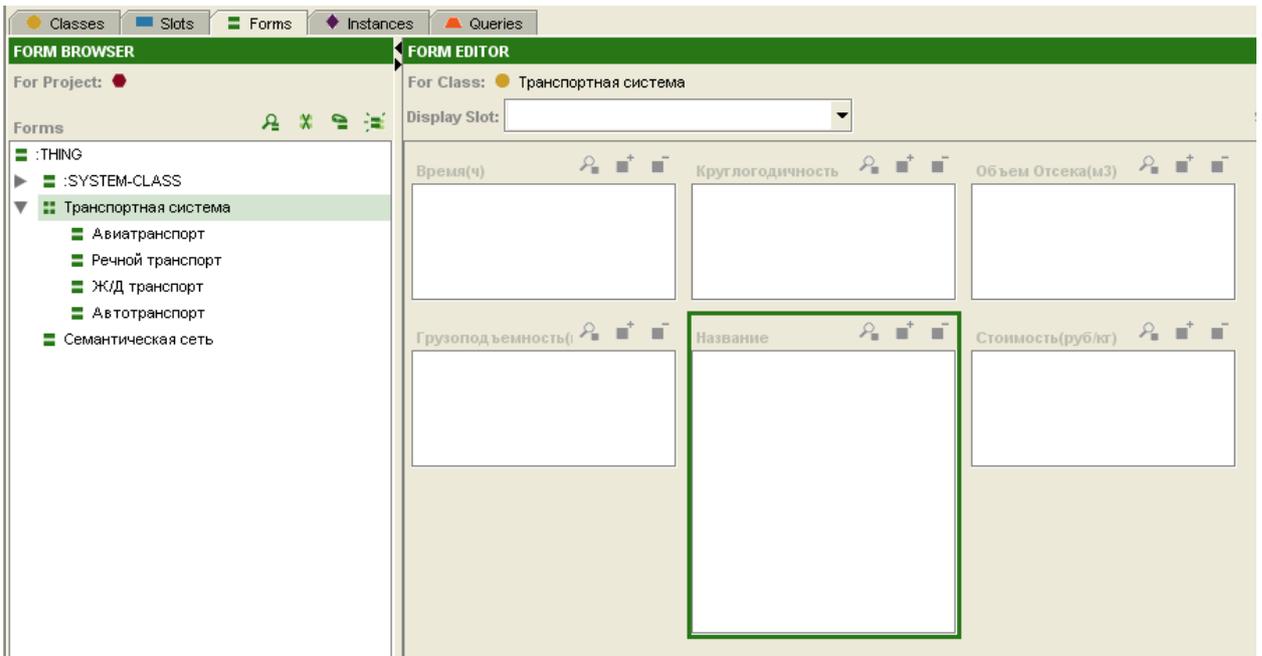


Рисунок 1.25 – Изменение размера виджета «Название»

Заметьте, что иконка перед формой класса «Транспортная система» изменилась с  на . Новая иконка указывает на то, что форма этого класса была изменена, и больше не является стандартной.

## Перемещение виджета

Для того чтобы изменить положение виджета нужно кликнуть на него мышью и при нажатой левой клавише мыши «перетащить» его в нужное место в окне форм (Forms) (Рисунок 1.26).

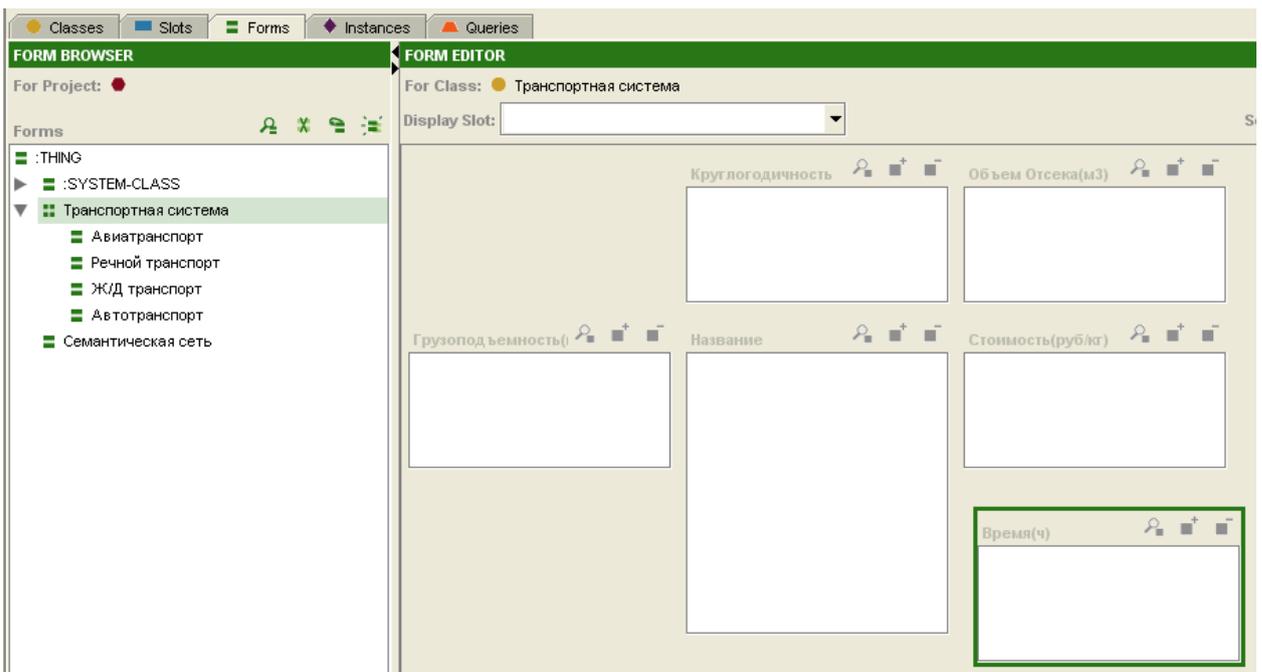


Рисунок 1.26 – Изменение положения виджета «Время»

Измените положение и форму виджетов в соответствии с рисунком 1.27.

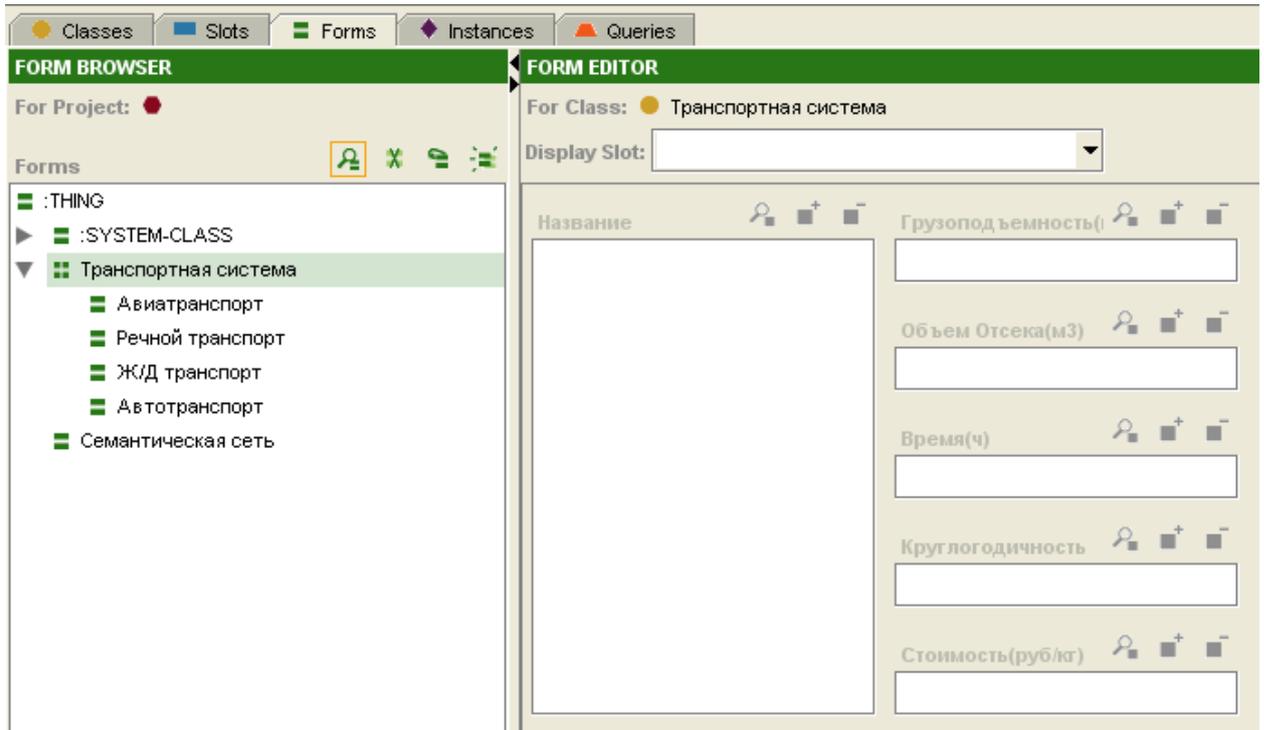


Рисунок 1.27 – Отредактированное окно форм

### Изменение кнопок виджета

Система Protege дает возможность настроить виджет так, чтобы он показывал другую надпись или другой список кнопок, нежели по умолчанию. Например, для того чтобы удалить кнопку , дважды кликните по виджету. В появившемся окне Configure в закладке Buttons уберите галочку в строке Show View Value Button (Рисунок 1.28) и нажмите ОК.

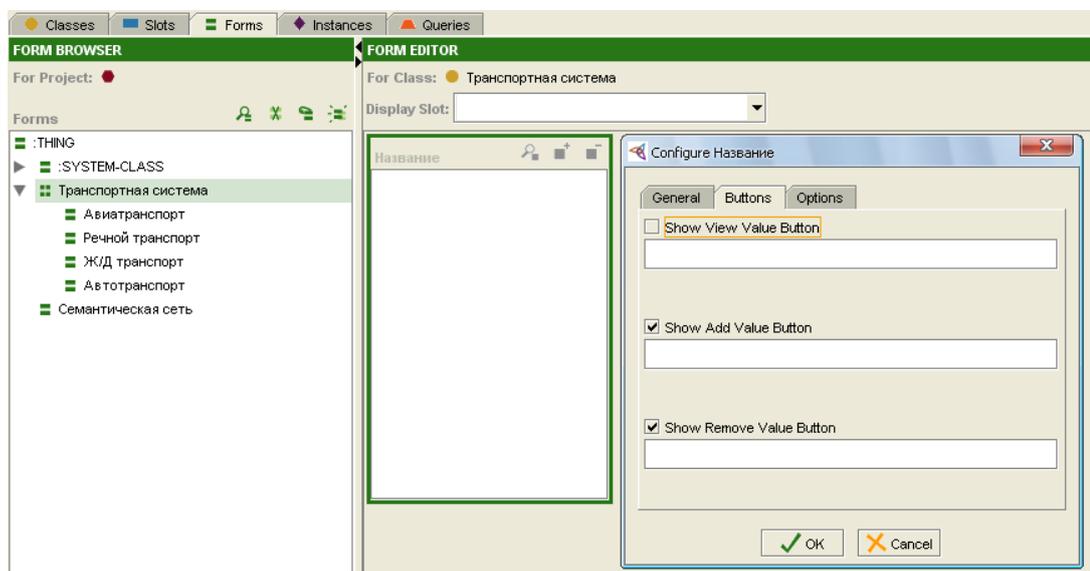


Рисунок 1.28 – Изменение количества кнопок виджета

После этого окно виджета изменится (Рисунок 1.29).

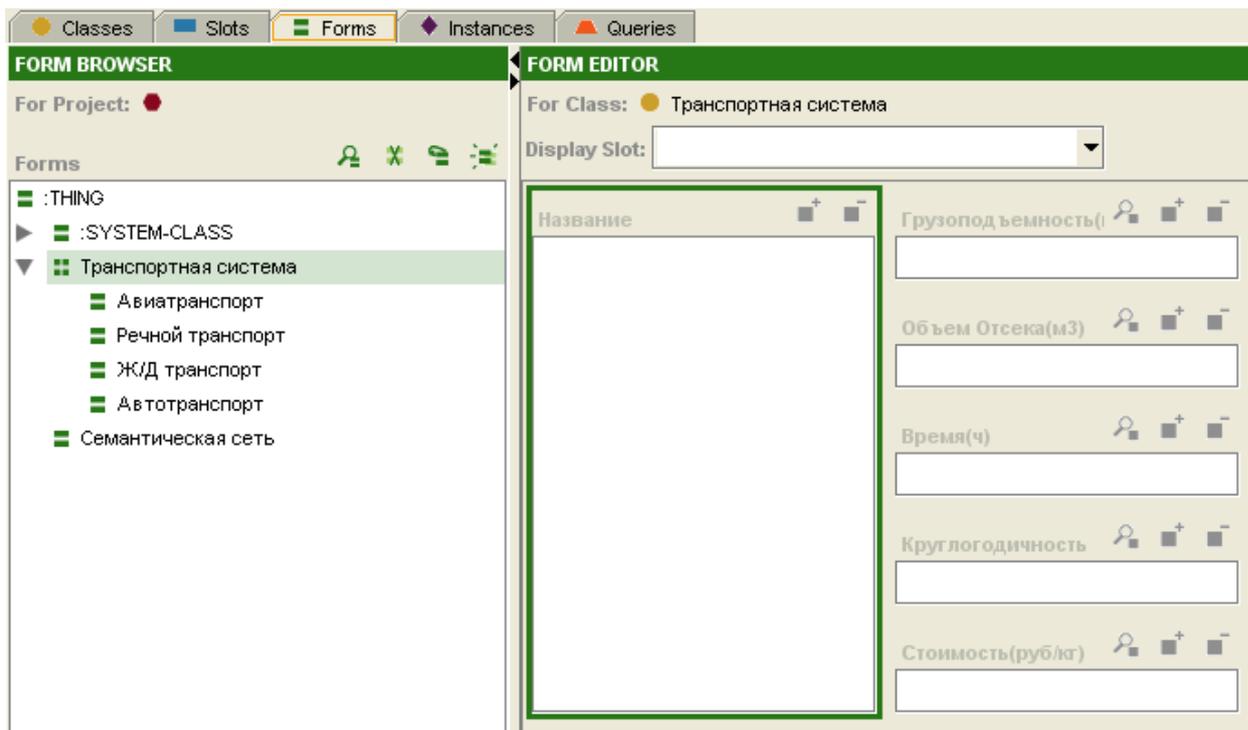


Рисунок 1.29 – Отредактированное окно виджета

### Изменение видимости виджета

Можно скрыть виджет так, чтобы он не был виден на форме и в редакторе экземпляров (при этом информация из онтологии не удаляется). Для того чтобы скрыть виджет для слота «название», кликните на него мышью и в правом верхнем углу в строке Selected Widget Type выберете <none> (Рисунок 1.30).

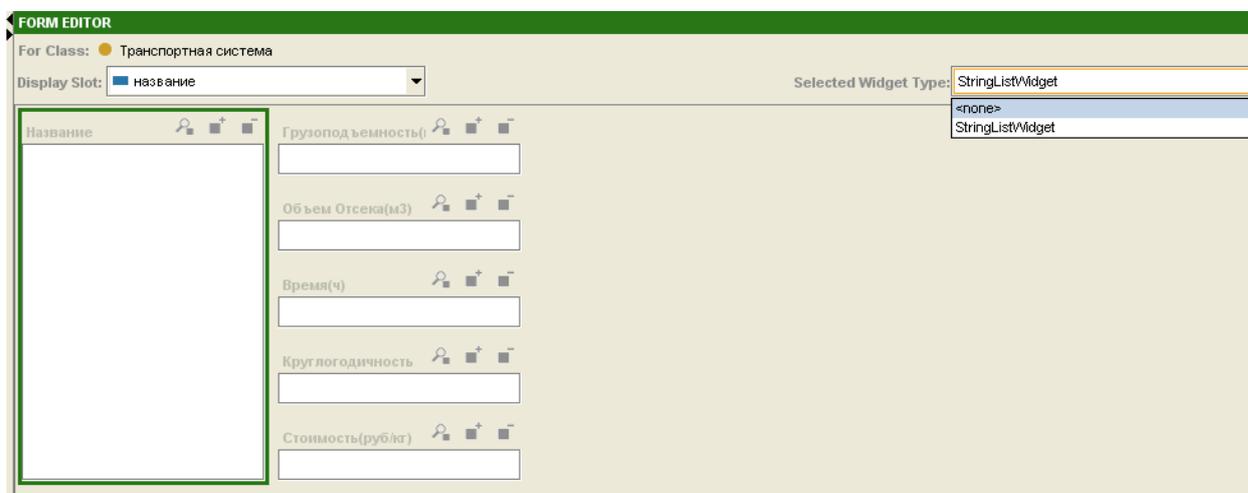


Рисунок 1.30 – Скрытие виджета

После этого виджет исчезнет из окна Form Editor (Рисунок 1.31).

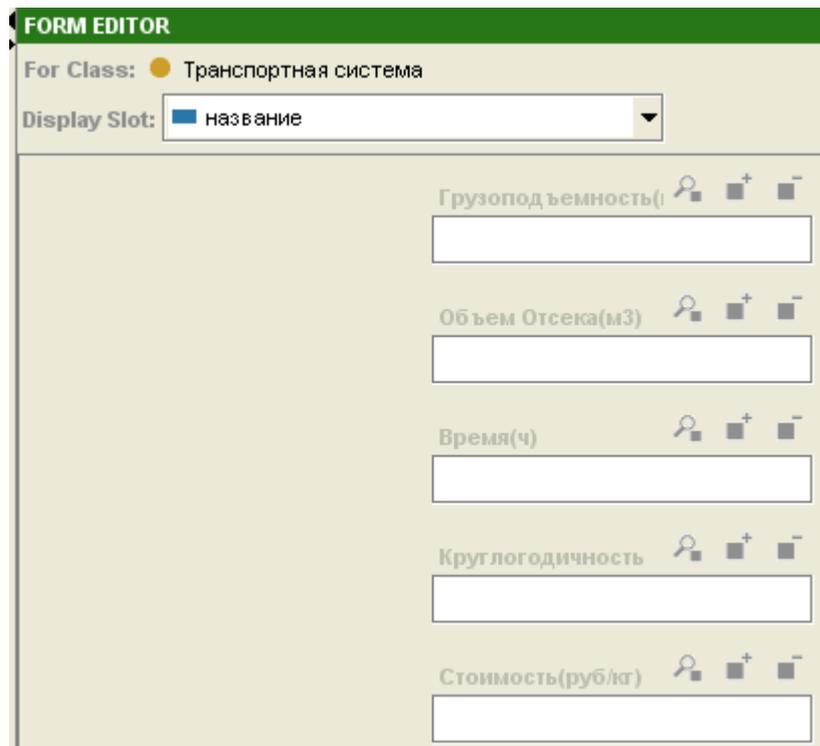


Рисунок 1.31 – Отредактированное окно Form Editor

### Отображение скрытого виджета

Для того чтобы отобразить скрытый виджет, нажмите на кнопку  View Form Customizations в окне Form Browser. В открывшемся окне виден список слотов и соответствующих им виджетов. Замените <none> на «String List Widget» и нажмите OK (Рисунок 1.32).

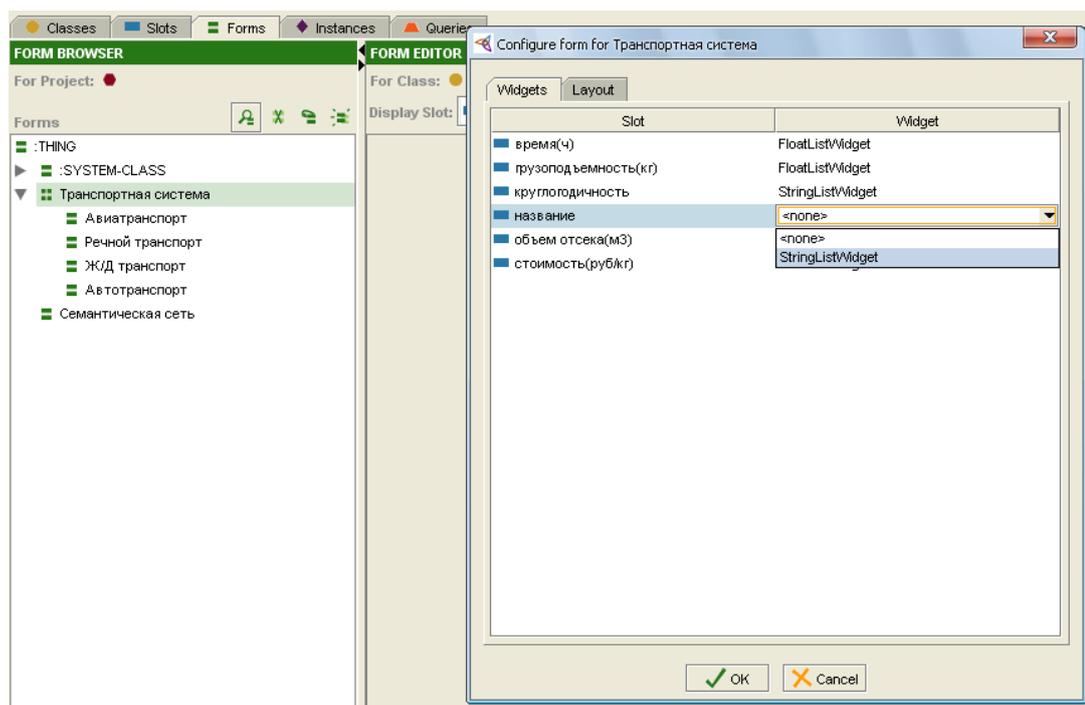


Рисунок 1.32 – Редактирование скрытого виджета

После этого виджет будет отображен в окне Form Browser в стандартной форме (Рисунок 1.33).

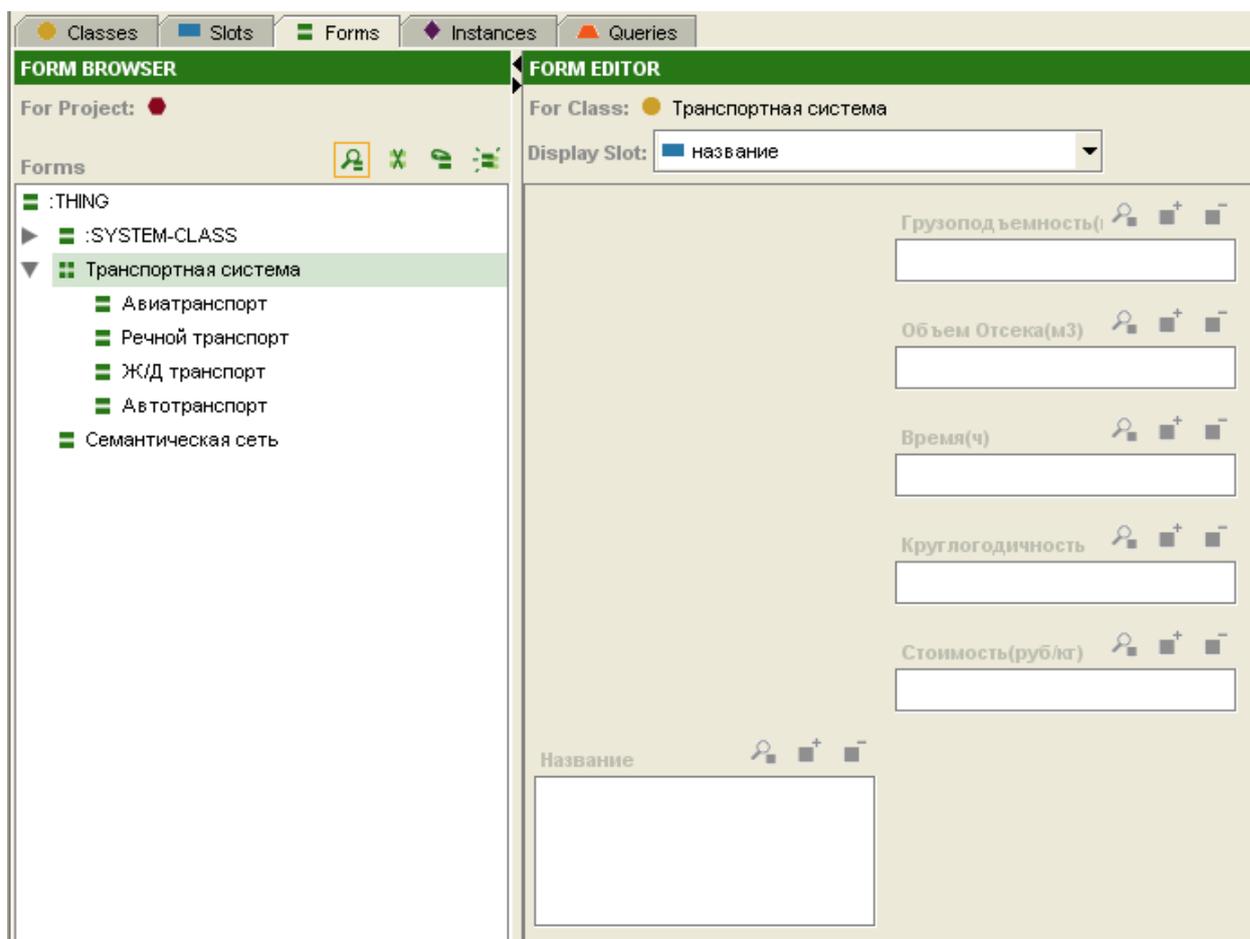


Рисунок 1.33 - Отображение скрытого виджета

Измените размер и положение виджета для подклассов класса «Транспортная система» в соответствии с рисунком 1.27. Для класса «Транспортная система» скройте все виджеты кроме виджета «название» (Рисунок 1.34).

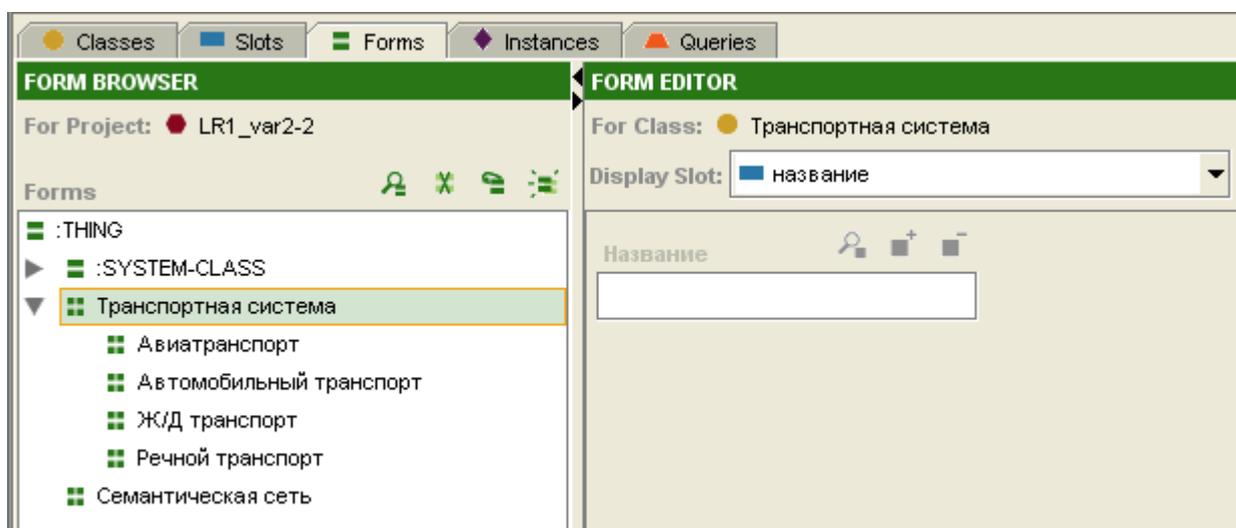


Рисунок 1.34 – Расположение виджета для класса «Транспортная система»

### 3.2.13. Создание экземпляров класса

Экземпляры классов – это данные создаваемой базы знаний. Перед вводом данных, необходимо окончательно проверить структуру проекта, потому что после их ввода изменение структуры может повлечь за собой потерю введенной информации. Кроме того, при добавлении новых слотов, необходимо заполнять их значения для старых экземпляров классов.

Для создания экземпляра класса «Авиатранспорт», переключитесь на закладку экземпляров (Instances), выберите «Авиатранспорт» и в окне Instances Browser нажмите кнопку  Create Instances, после этого в окне Instances Editor появятся слоты класса в той форме, которую Вы задали ранее. В окне Instances Editor в строке «Название» нажмите кнопку  Add Value, в появившемся окне Create String Value установите курсор в строку ввода, введите с клавиатуры название самолета «Ан-12» и нажмите ОК. (Рисунок 1.35).

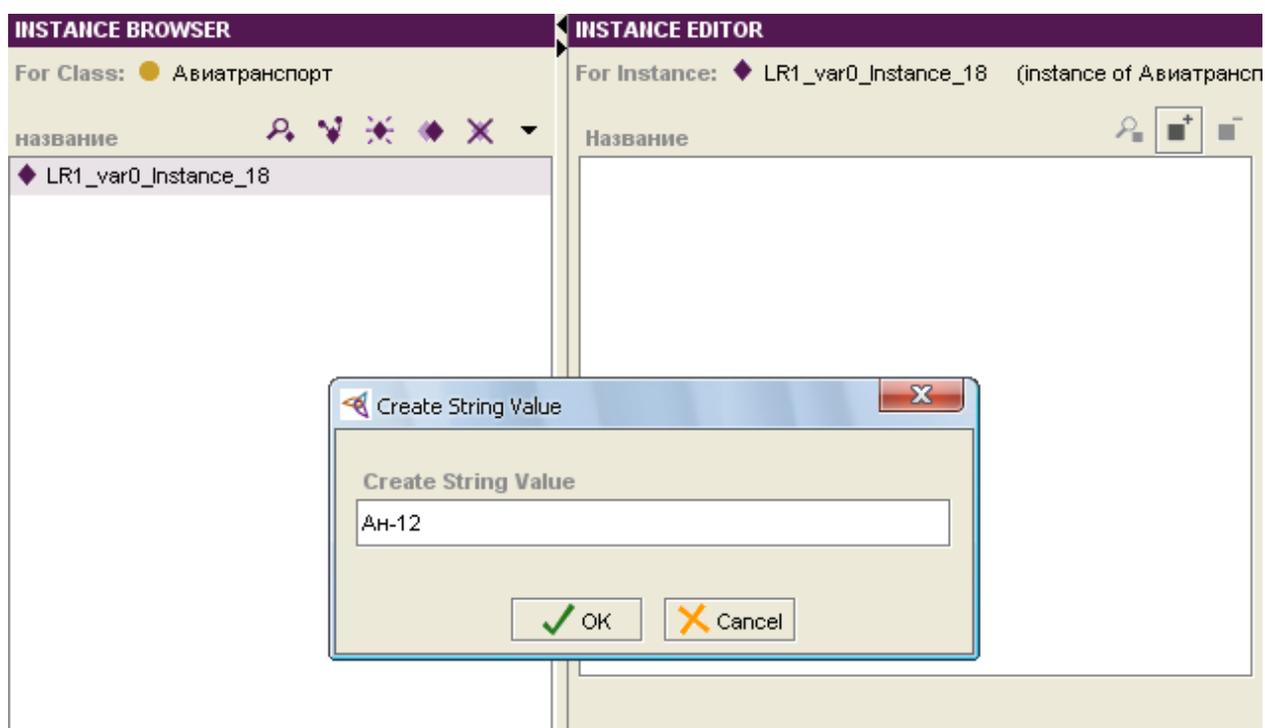


Рисунок 1.35 – Создание нового экземпляра

Аналогичным образом, нажимая на кнопку , в строках «Грузоподъемность», «Объем отсека», «Время» и т.д., введите данные (см. Таблица 1) для экземпляра «Ан-12». Обратите внимание, что имя в окне Instances Browser останется стандартным (Рисунок 1.36).

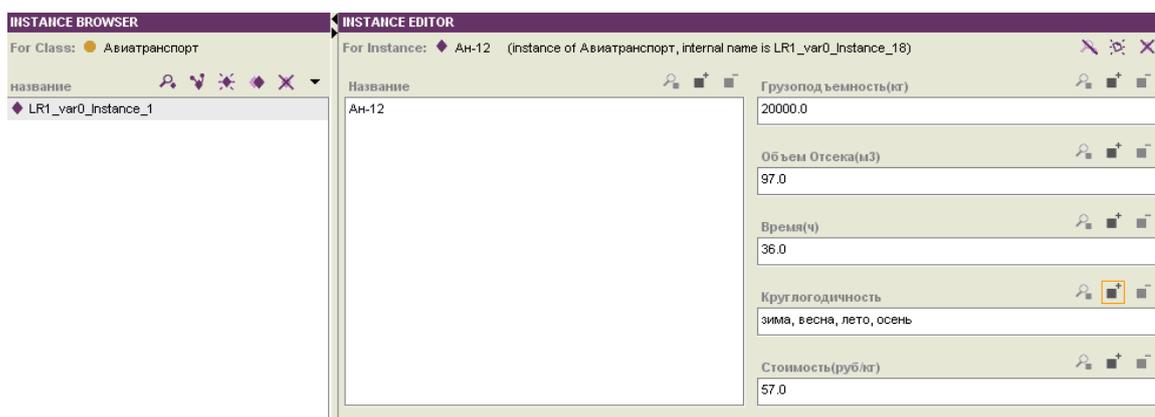


Рисунок 1.36 – Созданный экземпляр класса

В системе Protege, при попытке ввода значений, которые не удовлетворяют ограничениям слота, значения подсвечиваются красным цветом, то есть в поле, где выбран формат числа с плавающей запятой, нельзя вводить буквенные данные.

Таблица 1.1 Характеристики транспортных средств

Вид транспортного средства	Грузоподъемность (кг)	Объем отсека (м3)	Время (ч)	Круглогодичность	Максимальная стоимость (руб/кг)
<b>Авиатранспорт</b>					
Ан-12	20 000	97	36		57
Ан-24	5 500	110	24	зима, весна,	61
Ил-76	48 000	180	48	лето, осень	55
Як-76	2 700	51	24		63
<b>Ж/Д транспорт</b>					
Вагон 11-066	66 000	86	72		6
Вагон 11-217	68 000	104	72	зима, весна,	5.5
Вагон 11-К1	68 000	138	72	лето, осень	5
<b>Автотранспорт</b>					
Mercedes Actros	16 000	48	54		3.8
Газ 3302	1 500	9	36	зима, весна,	4.8
Зил 5301	3 000	18	48	лето, осень	4
<b>Речной транспорт</b>					
Волго-Дон	5 300 000	4 200	72	весна, лето,	4
Амур	2 500 000	1 500	68	осень	3.5

### 3.2.14. Установка слота отображения

Для каждого класса в онтологии, можно задать один или несколько слотов, слотами отображения. Система Protege будет показывать значение выбранных слотов, при каждом выводе экземпляра класса на экран. Если слот отображения не будет указан, то будет выведено стандартное имя, сгенерированное системой. Выбрать слот отображения можно еще до того как будут созданы экземпляры класса.

Для того чтобы выбрать слот отображения для класса «Авиатранспорт», из всплывающего меню окна Instances Browser выберете Set Display Slot и нажмите на «название» (Рисунок 1.37).

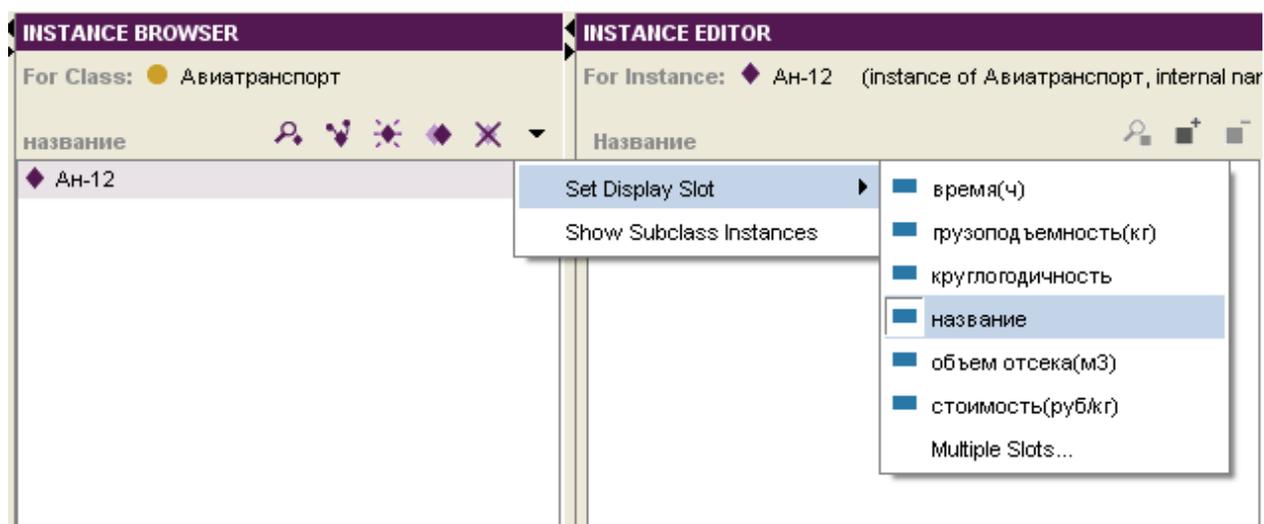


Рисунок 1.37 – Изменение стандартного имени экземпляра

После этого все создаваемые экземпляры этого класса будут отображаться в навигаторе экземпляров по названию в алфавитном порядке. Заметьте, что отображение для класса «Авиатранспорт» в панели иерархии классов (Class Hierarchy) изменилось после того, как был создан новый экземпляр класса. Единица в скобках означает, что этот класс имеет один экземпляр.

Аналогичным образом, используя данные таблицы 1.1, создайте экземпляры для классов «Авиатранспорт», «Автомобильный транспорт», «Ж/Д транспорт» и «Речной транспорт». Для этих классов также выберете слотом отображения слот «название».

### 3.2.15. Изменение параметров виджета для создания семантического представления онтологии

В пункте 1.1.2.10. был создан слот «Организационная структура» и слот «есть». В этом пункте будет рассказано, как с их помощью создать семантическое представление онтологии.

Перейдите в закладку Forms и в навигаторе форм выберите «Семантическая сеть». В окне Form Editor будут отображены виджеты в стандартной форме (Рисунок 1.38).

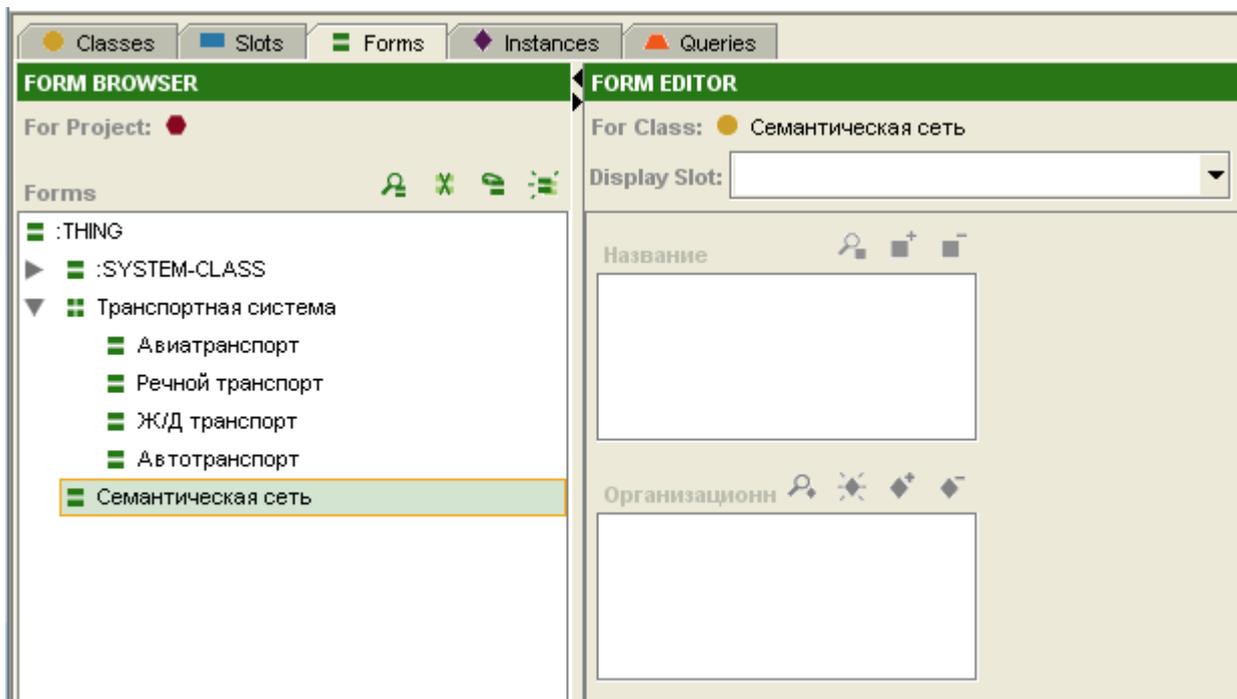


Рисунок 1.38 – стандартная форма виджетов класса «Семантическая сеть»

Выберете виджет «Организационная структура» для дальнейшего редактирования. В строке выбора типа виджета (Selected Widget Type) выберете графическое представление (Graph Widget) (Рисунок 1.39)

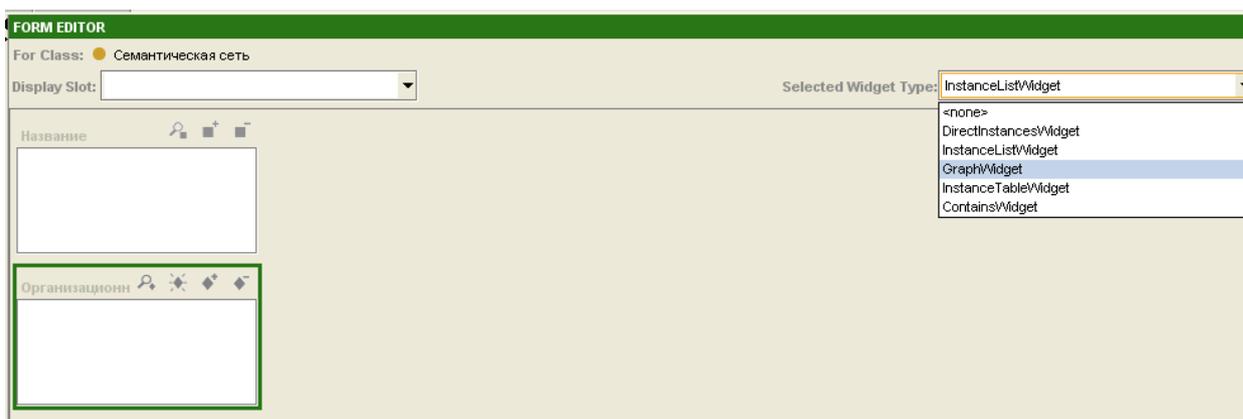


Рисунок 1.39 – Выбор типа виджета

После этого окно виджета будет представлено в виде графического поля с разметкой (Рисунок 1.40).

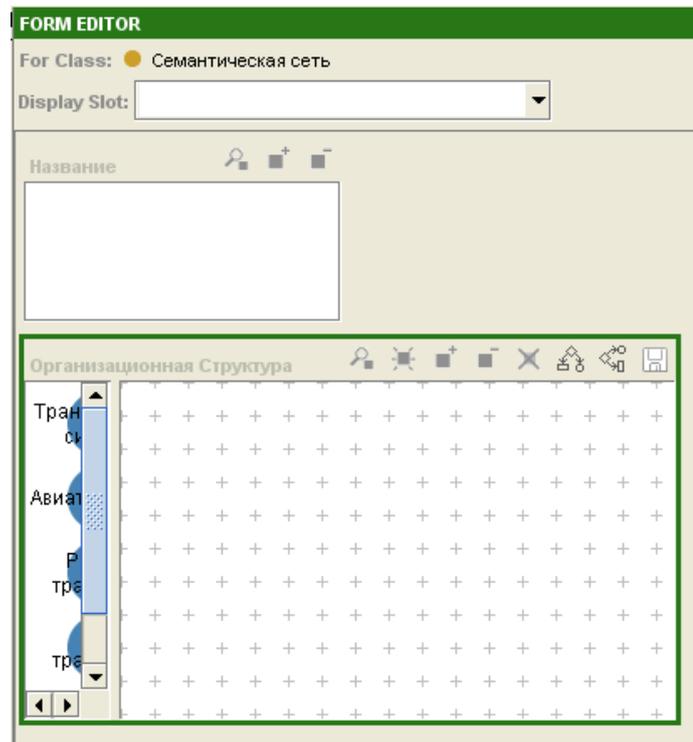


Рисунок 1.40 – Графическое представление виджета

Для того чтобы отредактировать этот виджет кликните по нему два раза и в окне Configure выберите закладку Nodes (Рисунок 1.41).

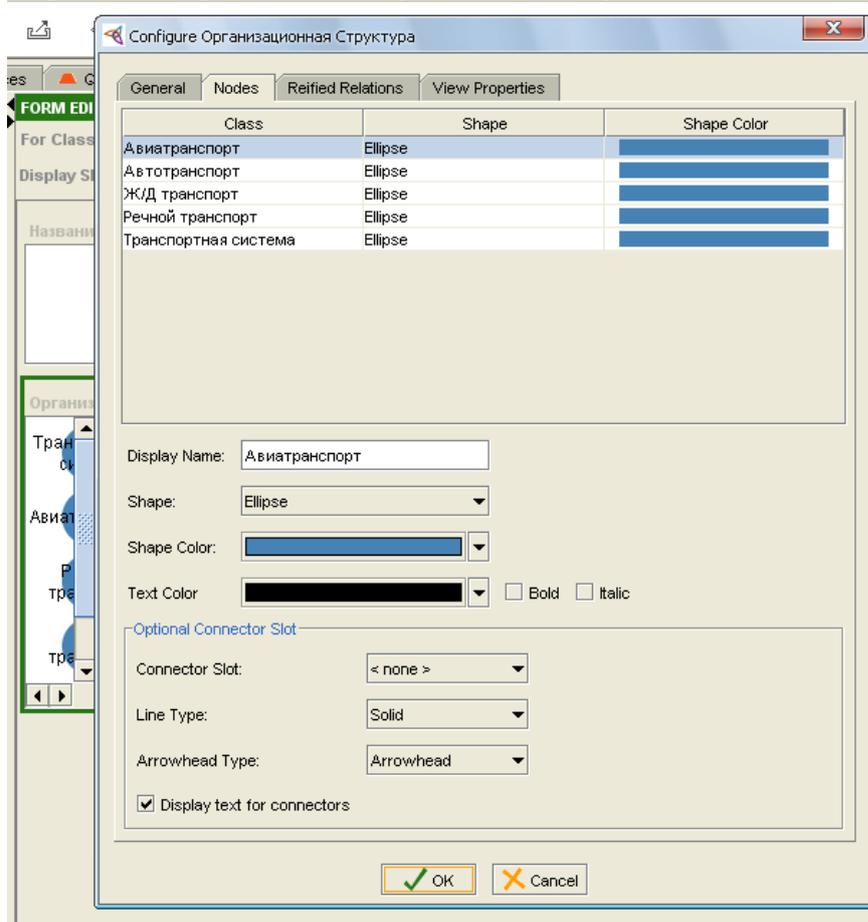


Рисунок 1.41 – Редактирование виджета

В открывшемся окне можно изменить настройки по умолчанию, а именно:

- форму и цвет значка, который будет отображаться в семантической сети,
- стиль и цвет текста надписи,
- тип связи между экземплярами.

Выберете класс «Транспортная система». Задайте для него форму значка Rectangle, выберите цвет и в строке Connector Slot выберите созданный ранее слот «есть» (Рисунок 1.42).

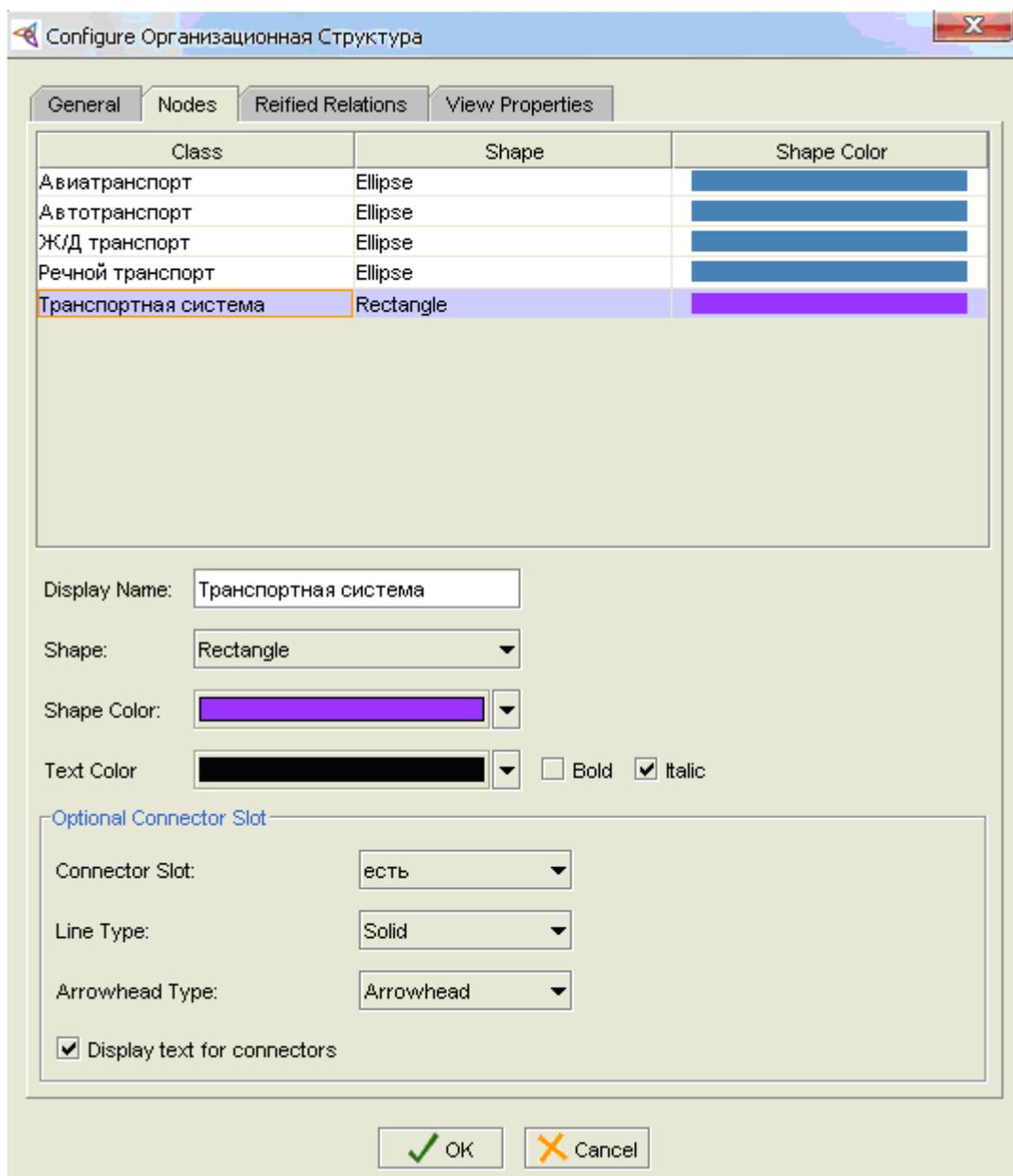


Рисунок 1.42 – Редактирование свойств значка «Транспортная система»

Далее для оставшихся классов задайте одинаковые свойства (кроме цвета значка). Форма – Octagon, Connector Slot – «есть» и нажмите ОК. Окончательный вид представлен на рисунке 1.43.

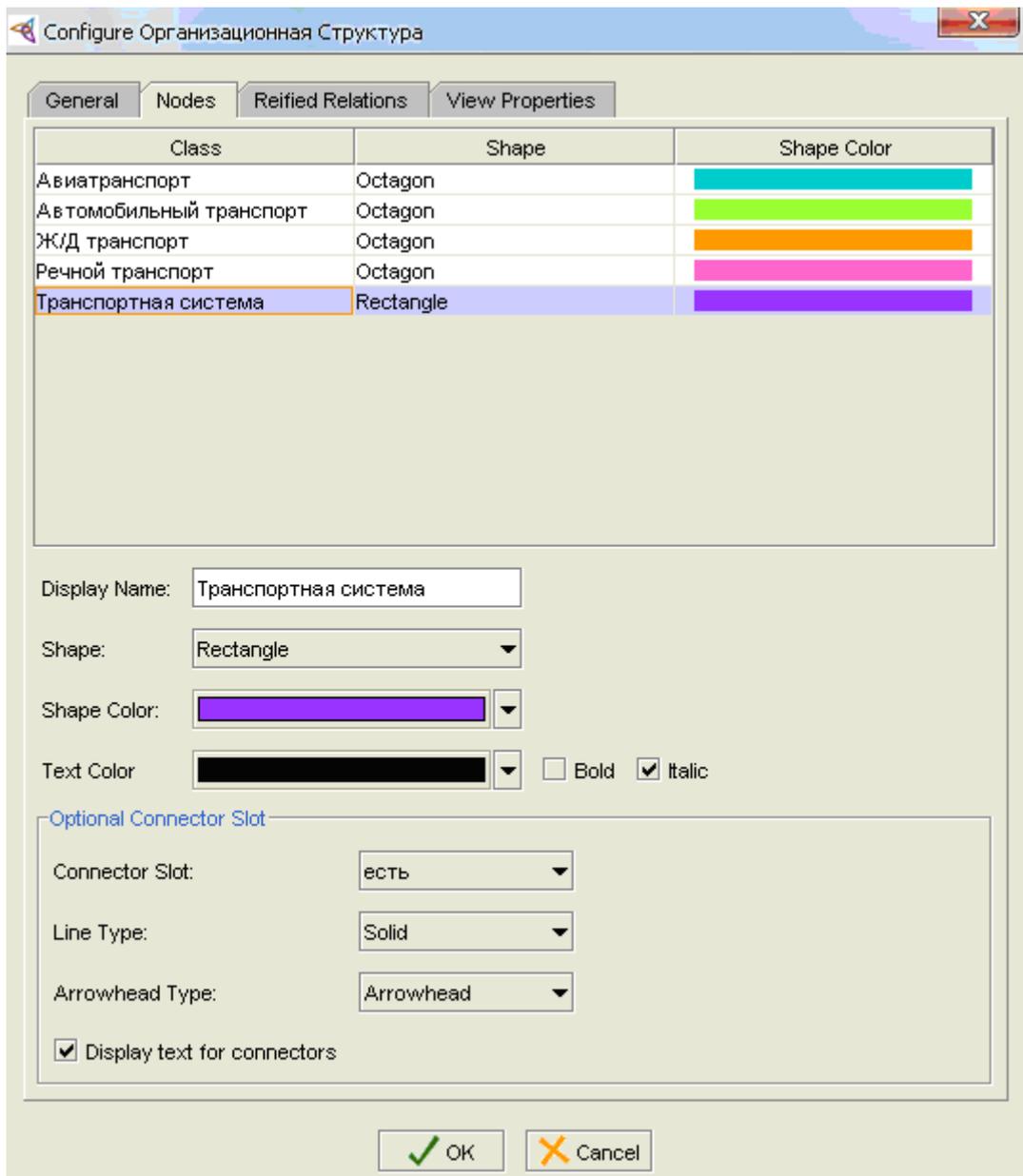


Рисунок 1.43 – Отредактированные свойства значков

Для лучшего представления семантической сети увеличьте размеры графического поля виджета (Рисунок 1.44).

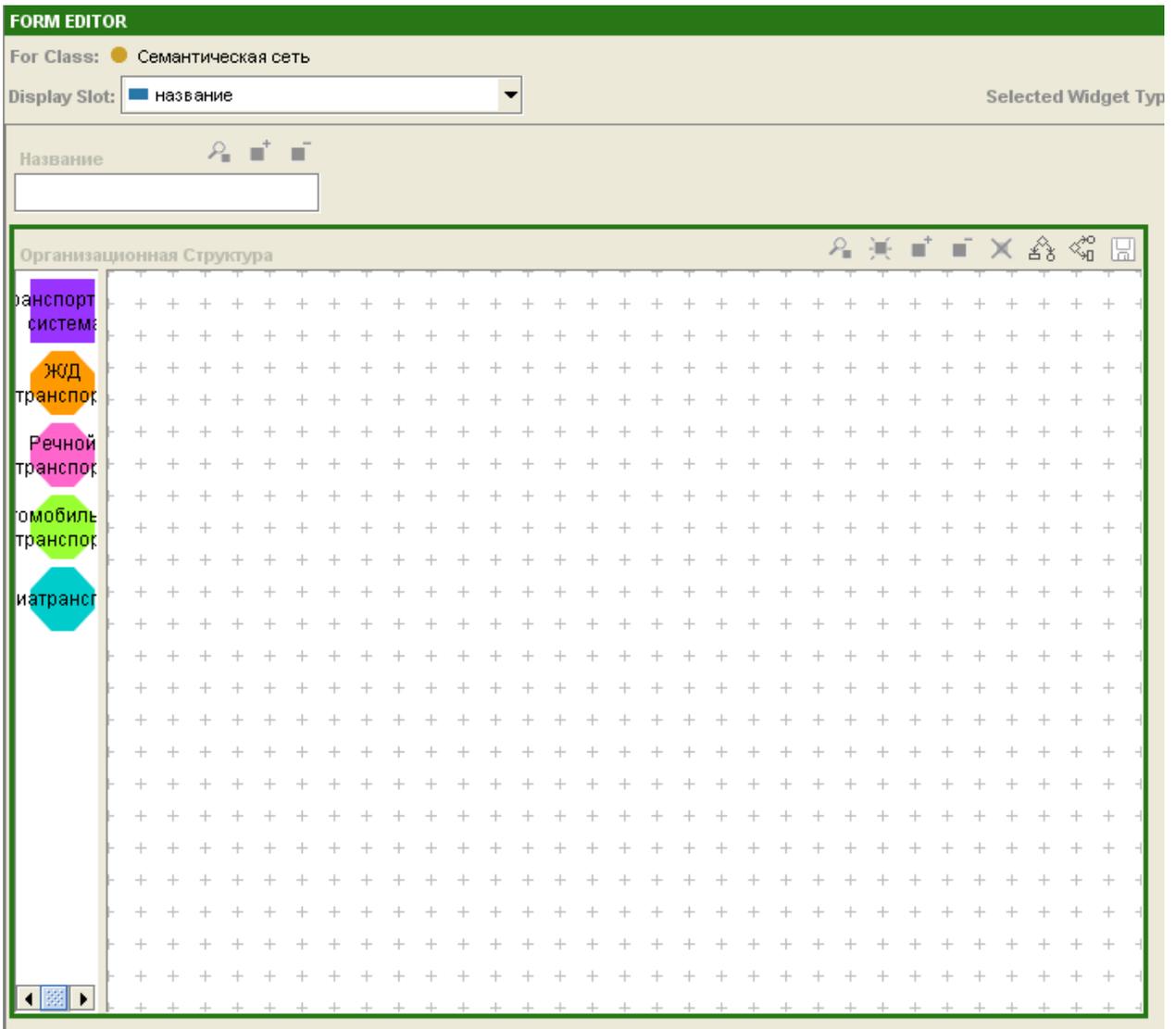


Рисунок 1.44 – Изменение размера виджета

### 3.2.16. Создание семантического представления онтологии

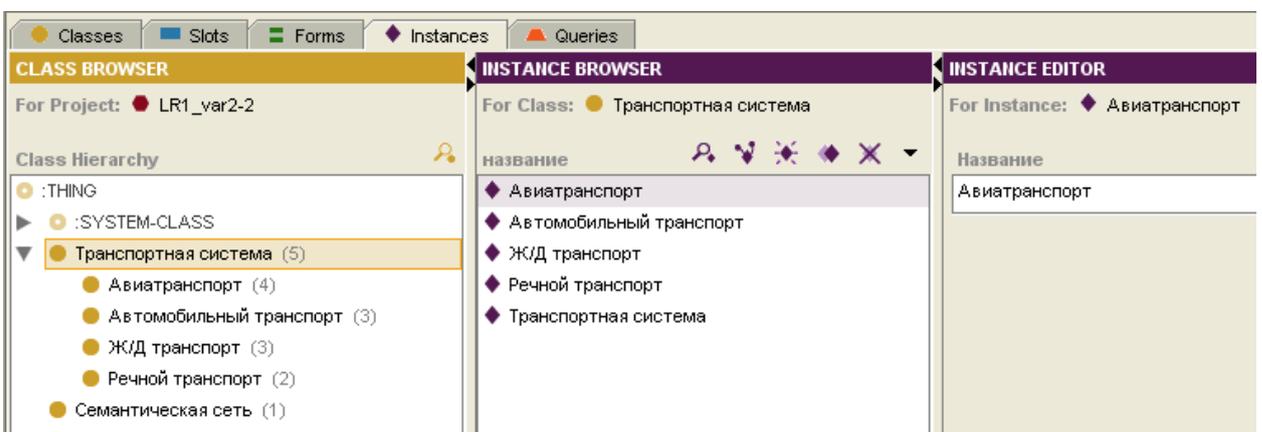


Рисунок 1.45 – Создание экземпляров класса «Транспортная система»

Перейдите в закладку экземпляров (Instances). Для класса «Транспортная система» создайте экземпляры «Транспортная система»,

«Авиатранспорт», «Автомобильный транспорт», «Ж/Д транспорт» и «Речной транспорт» (Рисунок 1.45).

Для создания семантической сети в окне Class Browser выберите класс «Семантическая сеть». Для этого класса создайте экземпляр с названием «Транспортная система» (Рисунок 1.46).

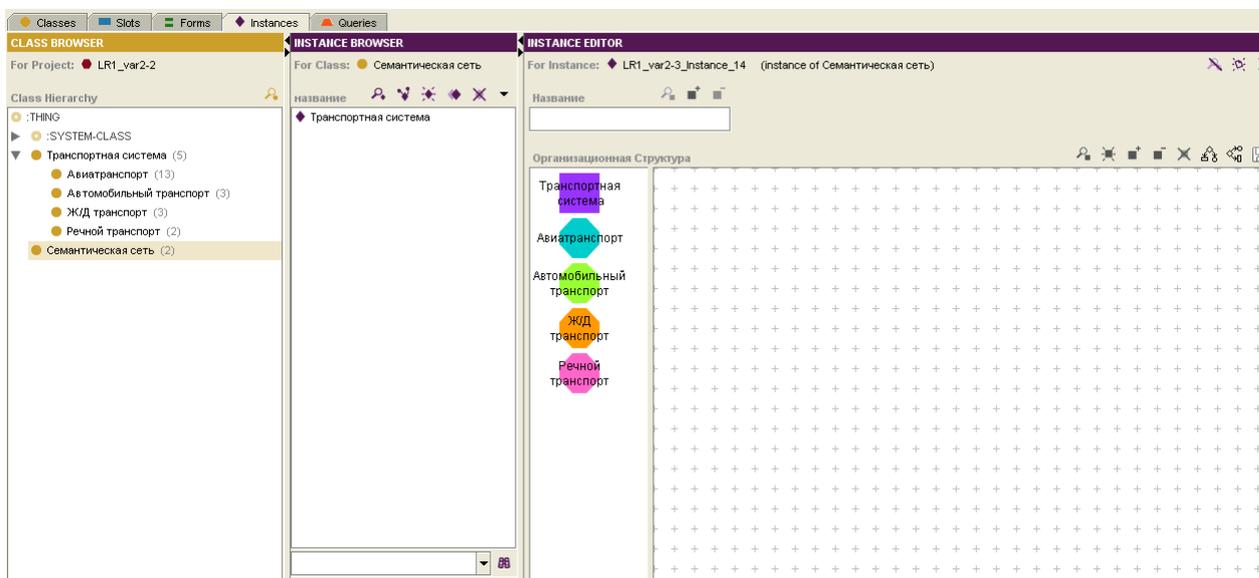


Рисунок 1.46 – Создание экземпляра класса «Семантическая сеть»

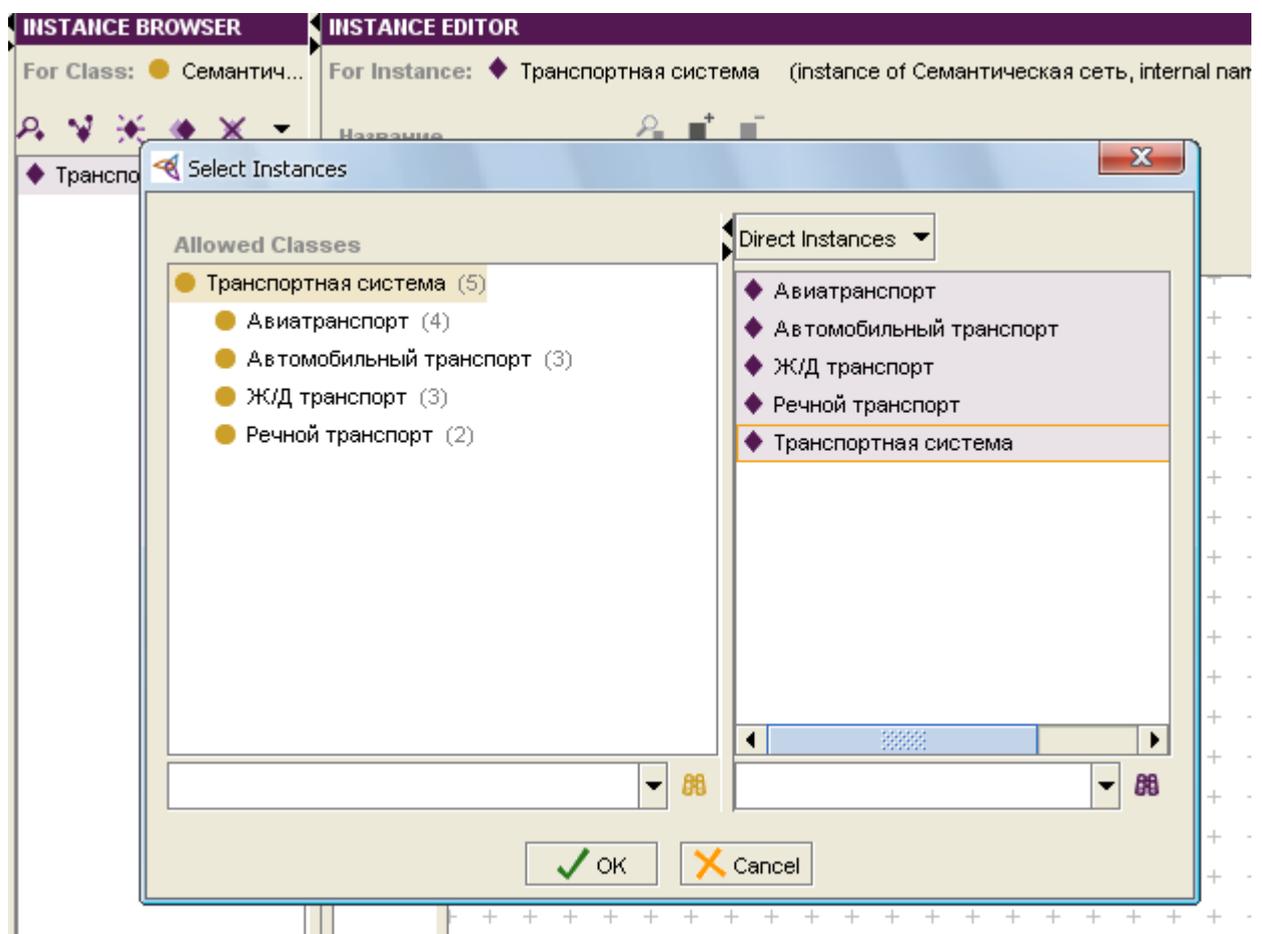


Рисунок 1.47 – Выбор экземпляров класса

После этого в окне Instance Editor появится графическое окно «Организационная структура». В правом верхнем углу нажмите на кнопку  Add Existing Instance. В появившемся окне слева выберете класс «Транспортная система», справа сочетанием клавиш Ctrl+A выберете все экземпляры этого класса и нажмите ОК (Рисунок 1.47).

В графическом окне, выбирая и удерживая нажатой левую кнопку мыши, расположите значки в соответствии с рисунком 1.48, а также измените размер значков исходя из размеров текста.

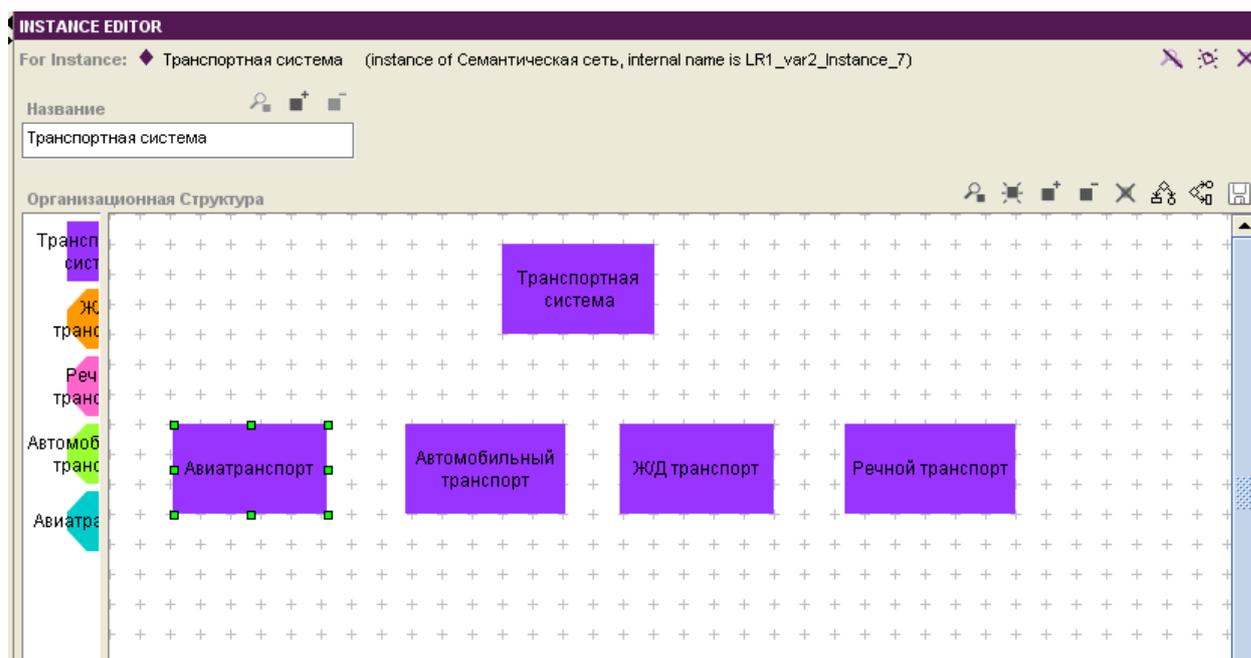


Рисунок 1.48 – Расположение значков в графическом окне

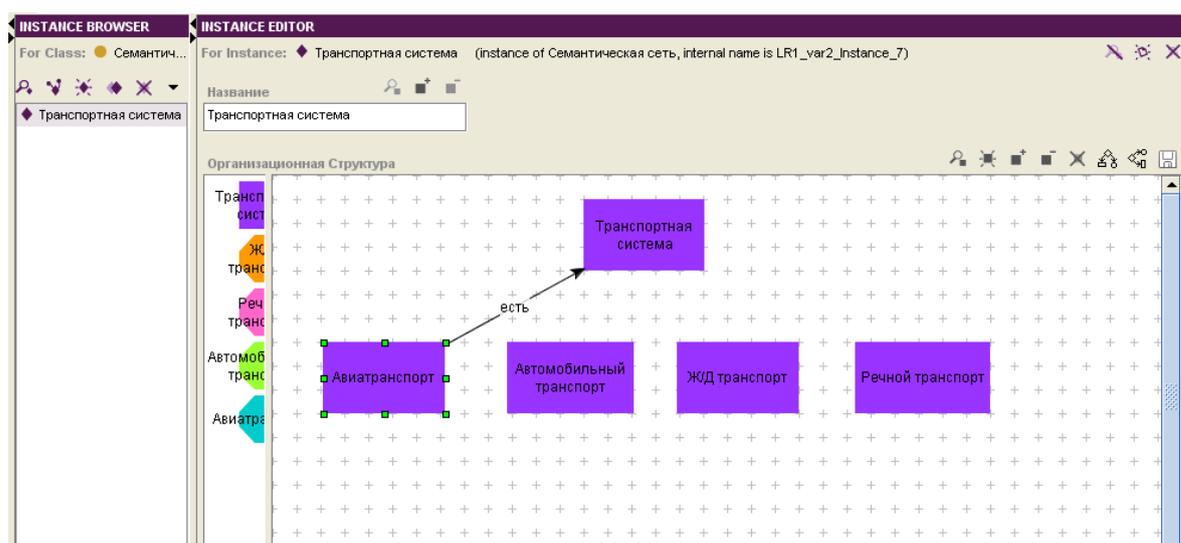


Рисунок 1.49 – Создание связи между элементами «Авиатранспорт» и «Транспортная система»

Для того, чтобы задать связи между элементами выберите значок «Авиатранспорт» после того как стрелка мыши примет вид руки, нажмите на значок и, удерживая нажатой левую кнопку мыши, укажите значок «Транспортная система». После того, как Вы убедитесь, что линия будет начерчена в нужном направлении, отпустите кнопку мыши. Будет создана связь между элементами с названием «есть», что буквально означает: «Авиатранспорт есть Транспортная система» (Рисунок 1.49).

Аналогичным образом задайте связи между элементами «Автомобильный транспорт», «Ж/Д транспорт», «Речной транспорт» и элементом «Транспортная система» (Рисунок 1.50).

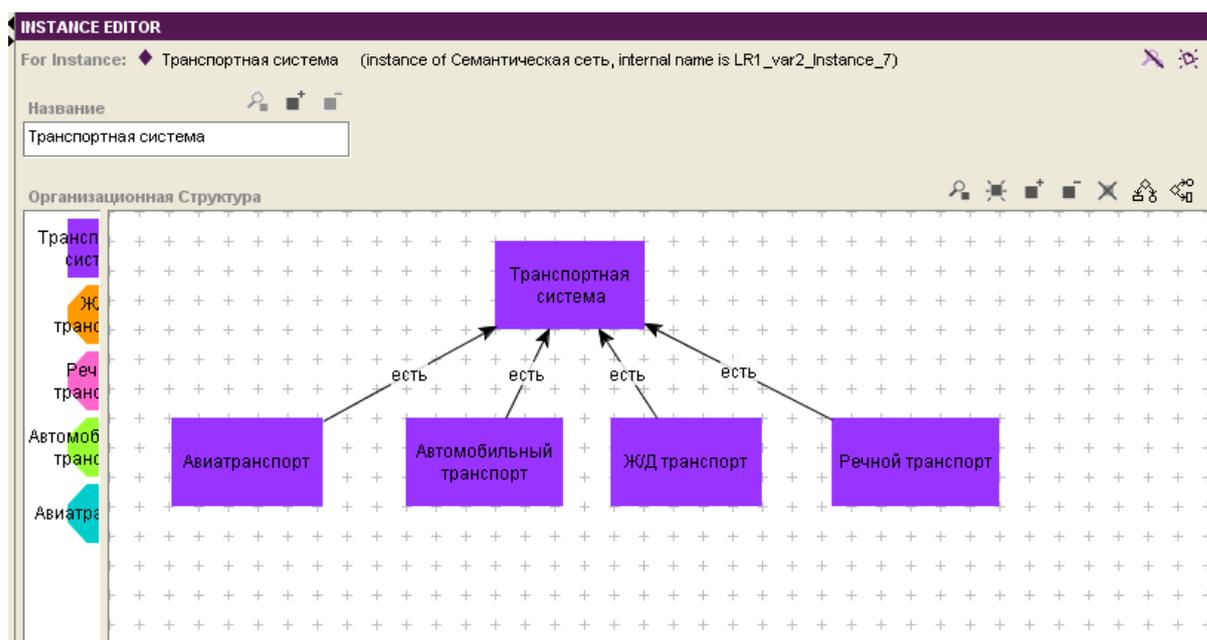


Рисунок 1.50 – Создание связей между элементами «Автомобильный транспорт», «Ж/Д транспорт», «Речной транспорт» и элементом «Транспортная система»

На следующем шаге необходимо включить в семантическую сеть экземпляры классов «Авиатранспорт», «Автомобильный транспорт», «Ж/Д транспорт» и «Речной транспорт». Это можно сделать описанным ранее способом. Семантическая сеть должна выглядеть в соответствии с рисунком 1.51.

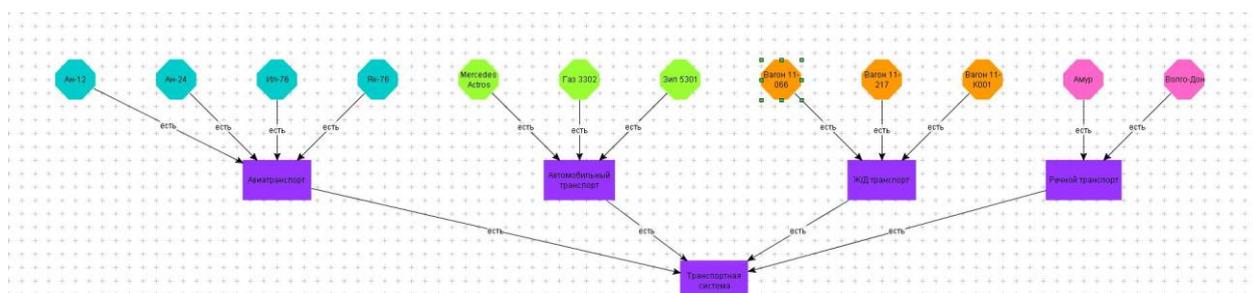


Рисунок 1.51 – Семантическая сеть

### 3.2.17. Создание запроса

После создания онтологии необходимо осуществить выбор транспортного средства по заданным параметрам. Рассмотрим создание запроса на следующем примере (Таблица 1.2).

Таблица 1.2 – Характеристики для создания запроса

Масса перевозимого груза (кг)	Объем перевозимого груза (м3)	Срок перевозк и груза (ч)	Время года для осуществления перевозки	Максимальная стоимость перевозки (руб/кг)
3 500	200	10-60	зима	62

В качестве начального и конечного пункта рассматриваются города Москва и Самара соответственно. Приоритетным параметром при решении этой задачи является стоимость перевозки груза. Поэтому «Стоимость» должна быть завершающим критерием, включаемым в запрос.

Для того чтобы создать запрос перейдите на закладку запросов (Queries). В строке Class нажмите кнопку  Select Class для выбора класса. В появившемся окне выберете класс «Транспортная система» (все его подклассы будут также включены в запрос) и нажмите ОК (Рисунок 1.52).

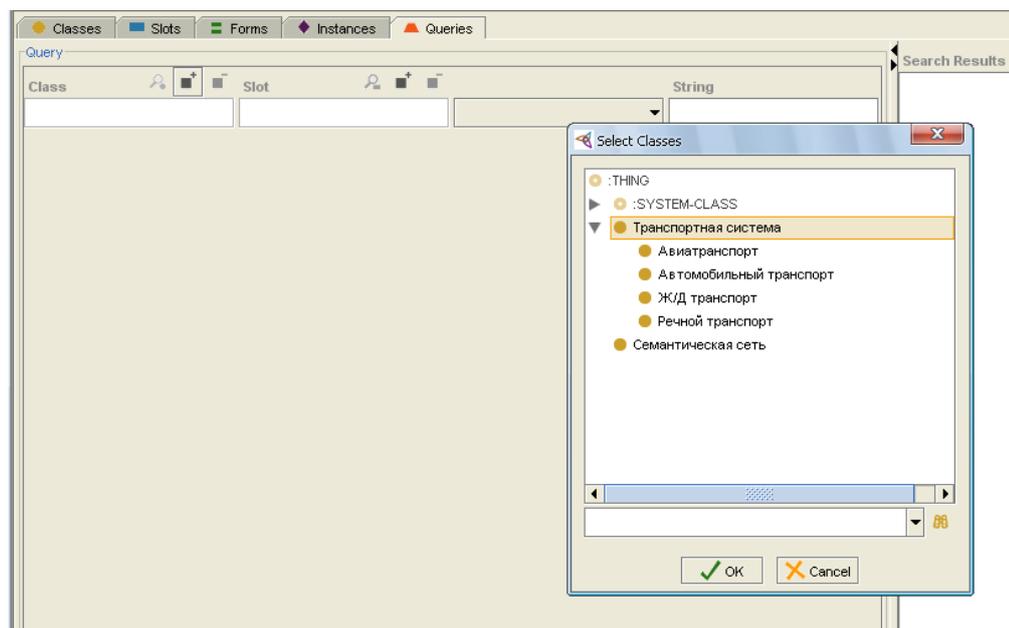


Рисунок 1.52 – Выбор класса для создания запроса

Для выбора слота в строке Slot нажмите кнопку  Select Slot. В появившемся окне (Рисунок 1.53) выберете «грузоподъемность» и нажмите ОК.

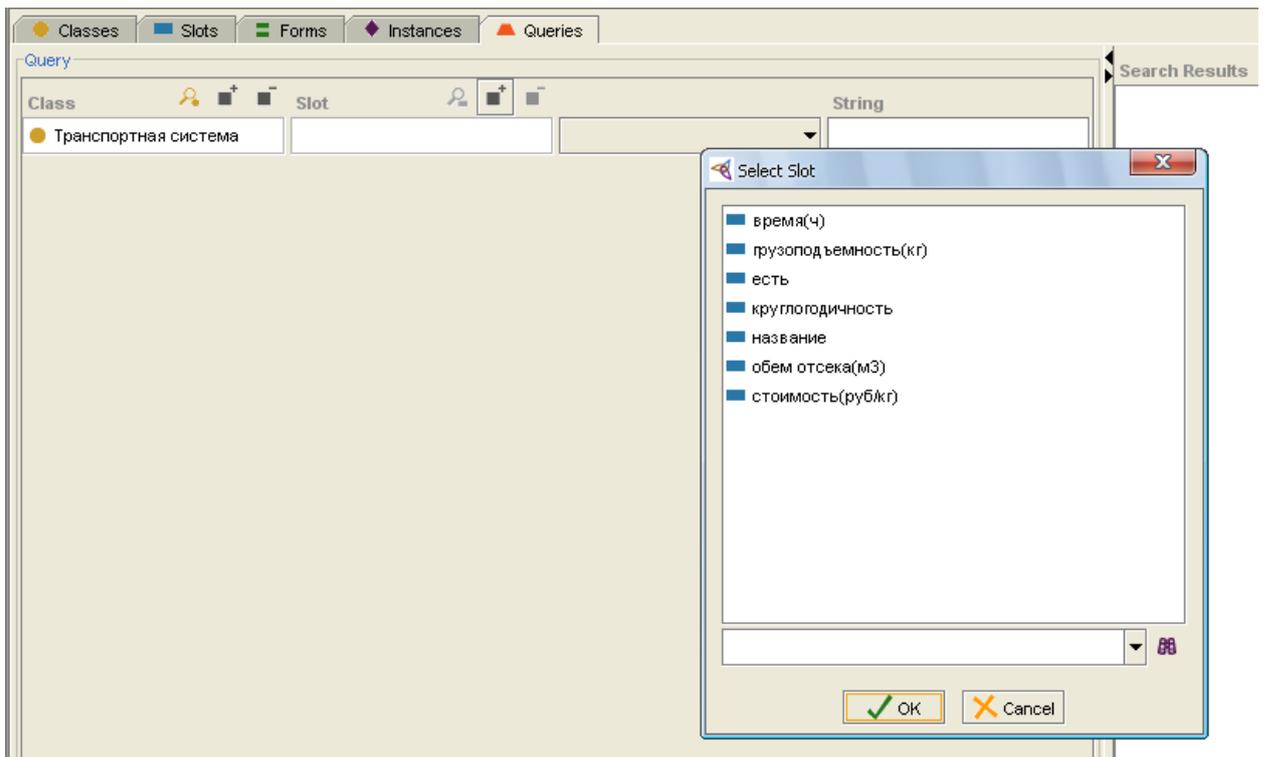


Рисунок 1.53 – Выбор слота для создания запроса

После выбора слота, меню справа от него стало активным и можно сделать выбор в соответствии с типом этого слота. В данном случае, тип Float – число с плавающей запятой (Рисунок 1.54).



Рисунок 1.54 – Меню запроса

В выпадающем списке можно выбрать:

- is greater than – больше чем,
- is less than – меньше чем,
- is – равно.

Выберете «is greater than» (больше чем) и в следующую строку введите значение 3500 (Рисунок 1.55).

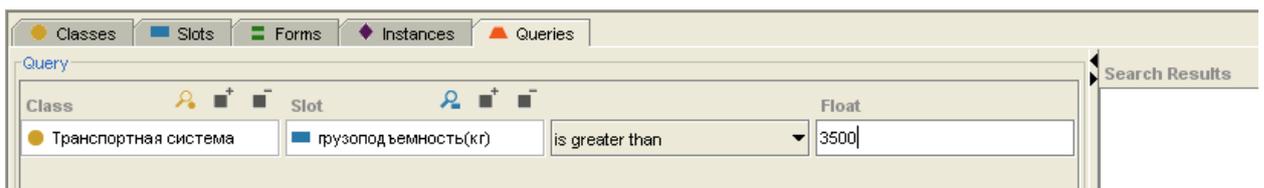


Рисунок 1.55 - Выбор параметров запроса

Для просмотра результата запроса нажмите кнопку «Find». В окне результатов запроса «Search Results» появится список транспортных средств способных перевозить груз массой более 3 500 кг (Рисунок 1.56).

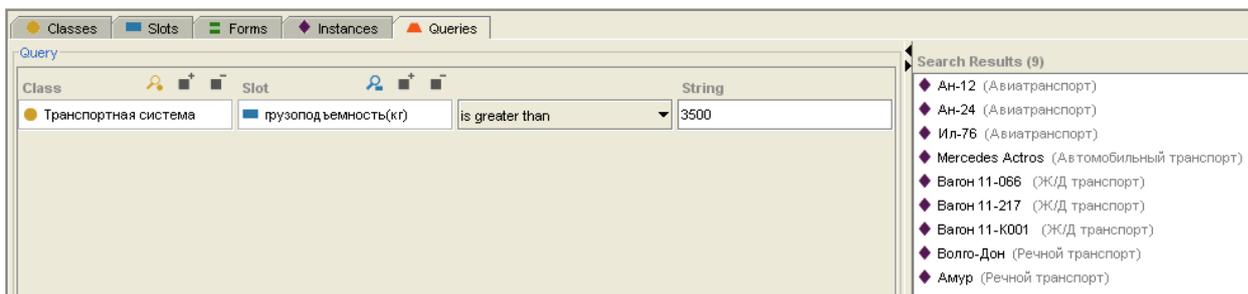


Рисунок 1.56 – Результат запроса по массе

Далее нажмите кнопку «More» для создания запроса по нескольким критериям. Появится дополнительная строка запроса (Рисунок 1.57).

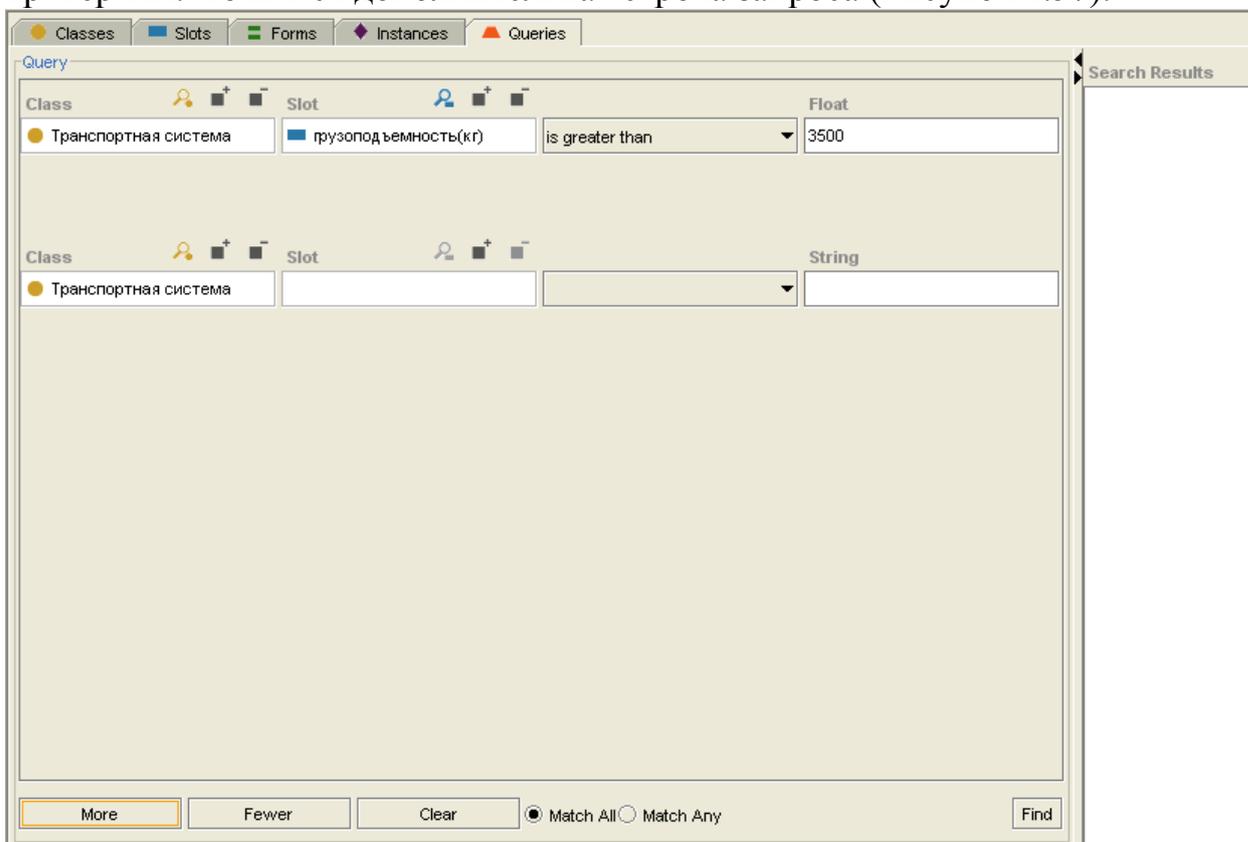


Рисунок 1.57 – Добавление строки запроса

Выберете слот «Круглогодичность». Этот слот имеет тип «String» поэтому выберете в меню «Contains» и введите с помощью клавиатуры - «зима». Результат запроса изменится, из него будут исключены транспортные средства, не осуществляющие перевозку грузов в зимнее время года (Рисунок 1.58).

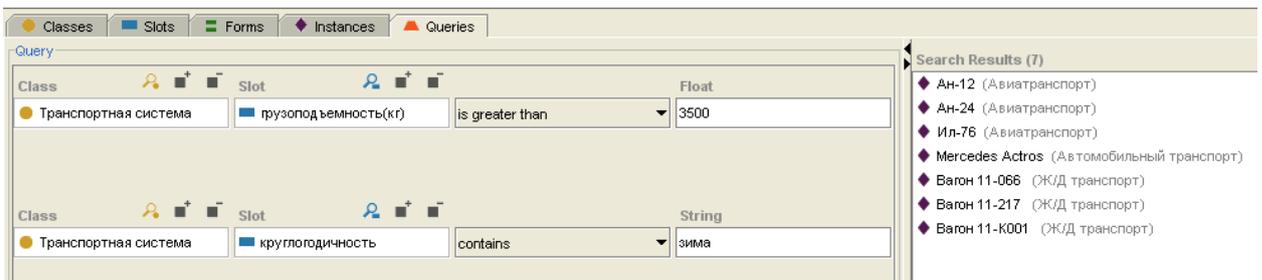


Рисунок 1.58 – Добавление параметров запроса

Выберете слот «Время», задайте значение «больше чем 10». Создайте еще одну строку со значением «меньше чем 60». Нажмите «Find». Результат запроса сузится с 7 до 4 экземпляров, так как был исключен транспорт, осуществляющий перевозку более чем за 60 часов (Рисунок 1.59).

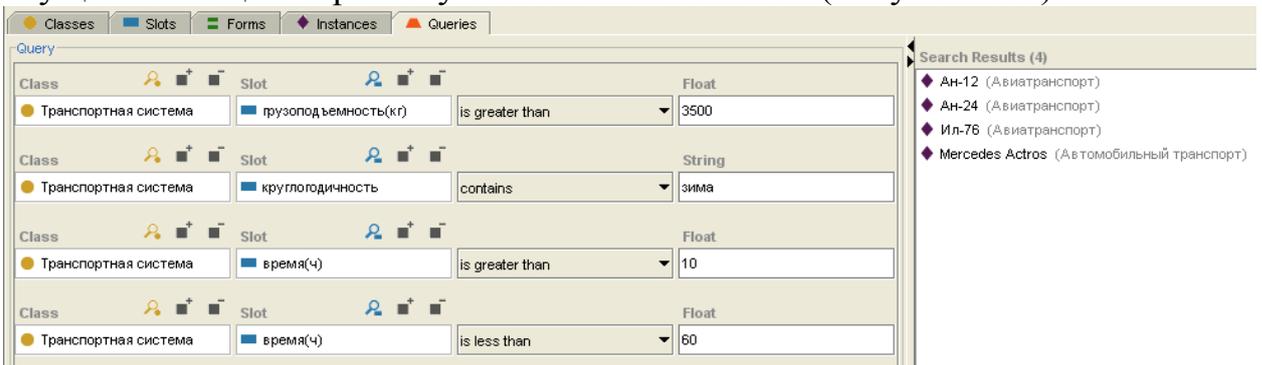


Рисунок 1.59 – Результат запроса с учетом времени перевозки

Выберете слот «Объем отсека» и задайте ограничения по объему. Нажмите «Find». Результат запроса равен нулю, это означает, что ни одно из транспортных средств удовлетворяющее предыдущим параметрам не сможет перевезти груз объемом больше 200м3 (Рисунок 1.60).

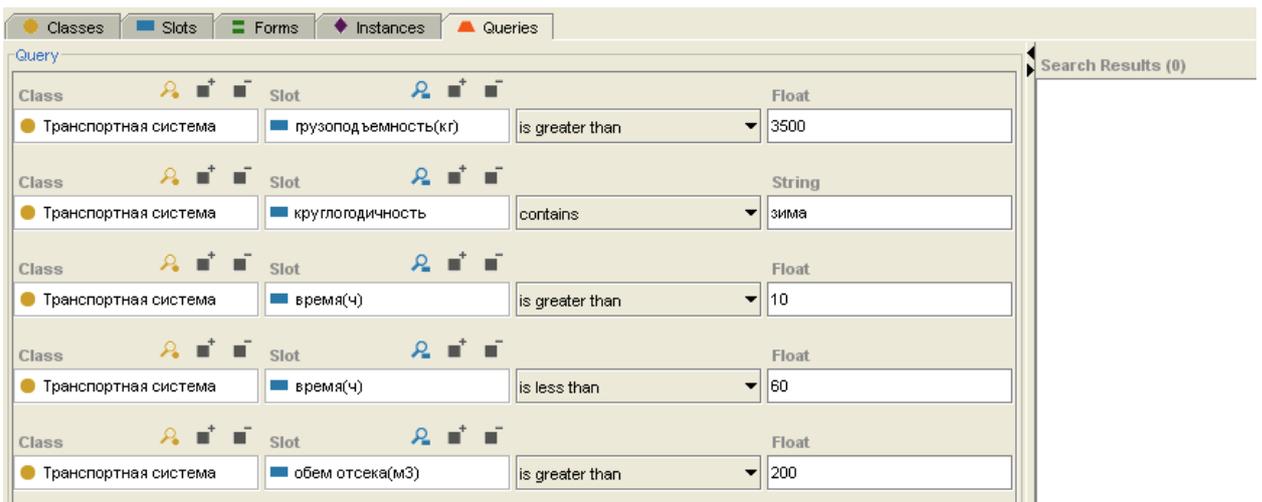


Рисунок 1.60 – Результат запроса с учетом объема перевозимого груза

В соответствии с допущениями задания (см. глава 1.1.3, п.4) груз можно разделить на две, три или четыре части. Разделите груз на две части и измените данные в столбцах «грузоподъемность» и «объем отсека». В

результате нового запроса будут получены два транспортных средства способных перевезти груз объемом более 100м3 (Рисунок 1.61).

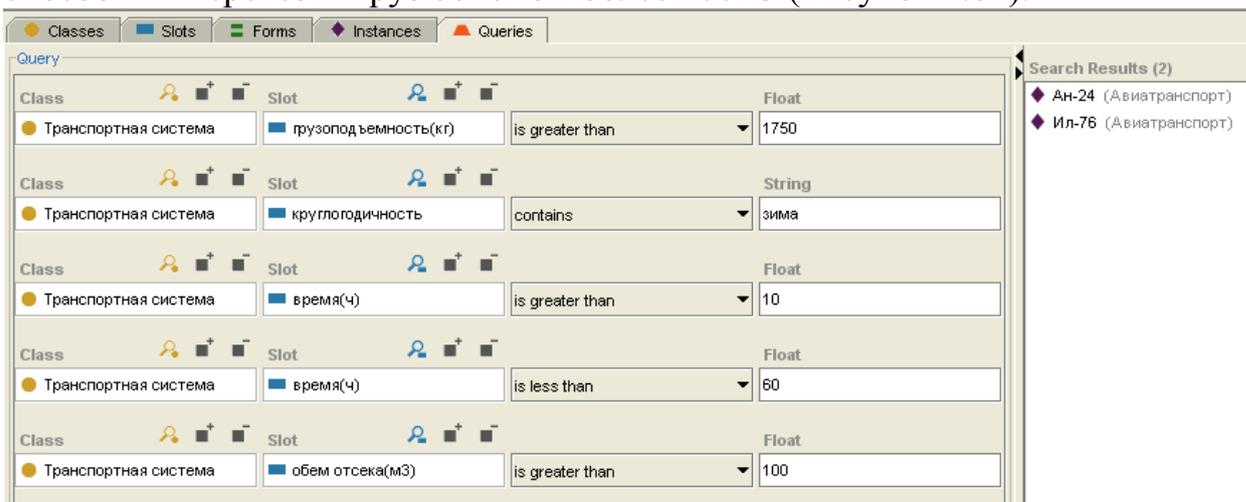


Рисунок 1.61 – Результат измененного запроса

После того, как стало известно, что груз будет разделен на две части и перевезен с помощью авиатранспорта, включите в запрос параметр стоимости. В строке «Float» введите «не больше 62». Результат запроса не изменился, то есть, оба транспортных средства могут осуществить перевозку груза заданной массы и объема в требуемые сроки в соответствии со временем года (Рисунок 1.62).

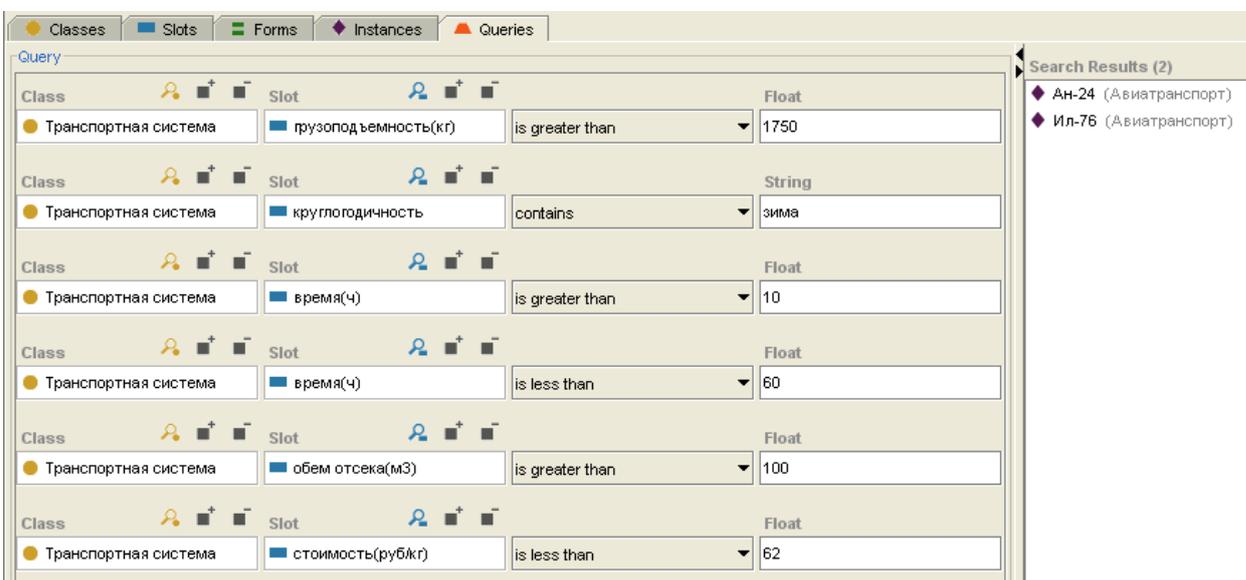


Рисунок 1.62 – Окончательный вид запроса

На следующем этапе необходимо выбрать оптимальное транспортное средство по критерию стоимости.

Для того чтобы увидеть свойства каждого транспортного средства, в том числе и стоимость перевозки, поочередно кликните двойным щелчком мыши в окне результатов запроса на «Ан-24» и «Ил-76» (Рисунок 1.63).

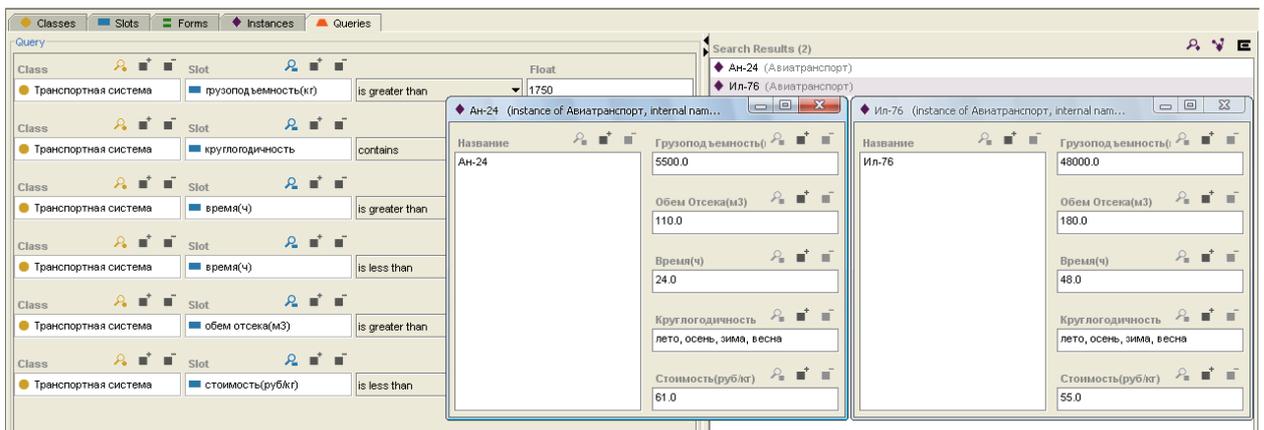


Рисунок 1.63 – Окна свойств экземпляров

В результате сравнения стоимости перевозки, можно сделать выбор в пользу самолета Ил-76.

### 3.2.18. Сохранение запроса

Protégé дает возможность сохранять запрос в библиотеке до или после того, как он будет запущен. Для этого нажмите «Add to Query Library», в появившемся окне введите имя запроса «Запрос№1» и нажмите ОК (Рисунок 1.64).

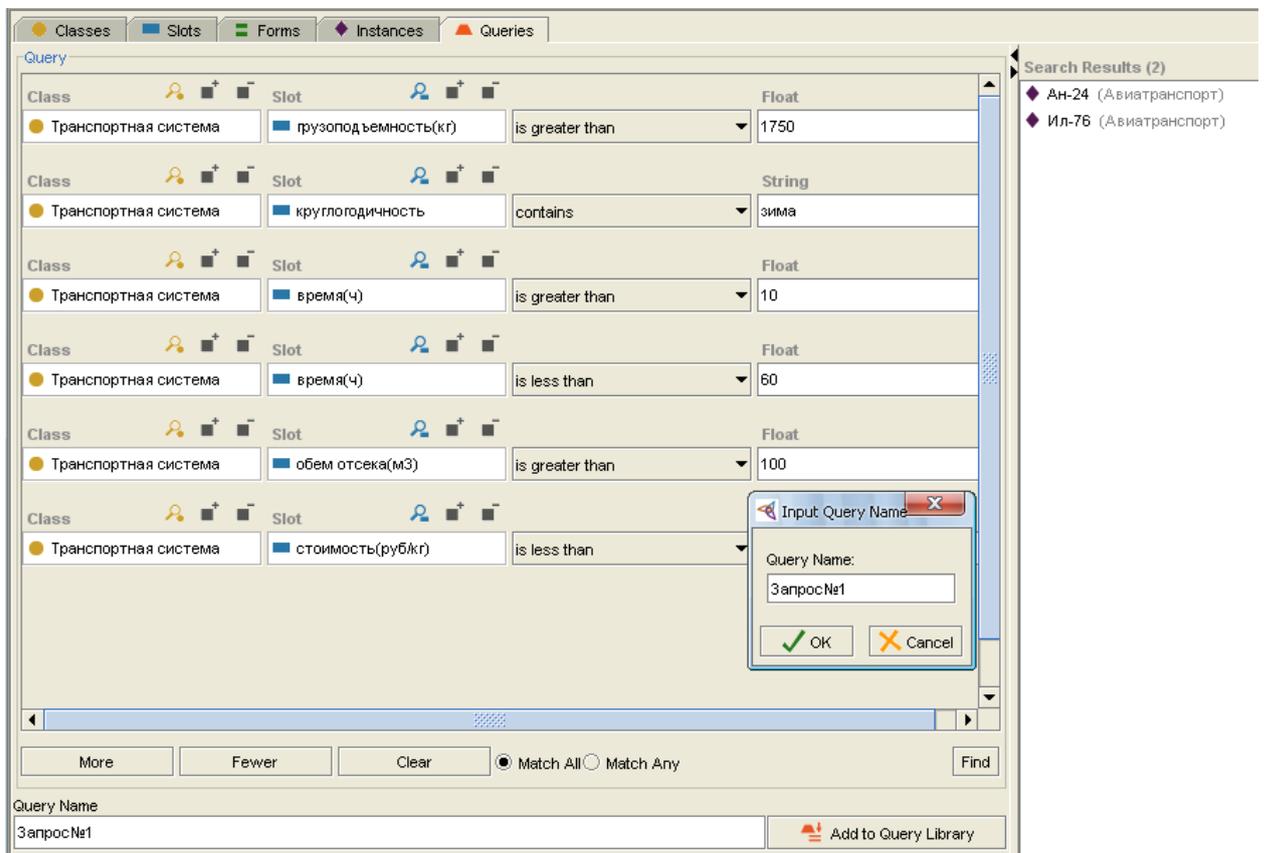


Рисунок 1.64 – Сохранение запроса

### 3.2.19. Загрузка запроса

Прежде чем загрузить сохраненный запрос, очистите поле запросов нажав на «Clear». В библиотеке запросов (Query Library) выберете «Запрос№1» и нажмите на кнопку  (Retrieve Query). Параметры запроса появятся в окне запросов «Query» (Рисунок 1.65).

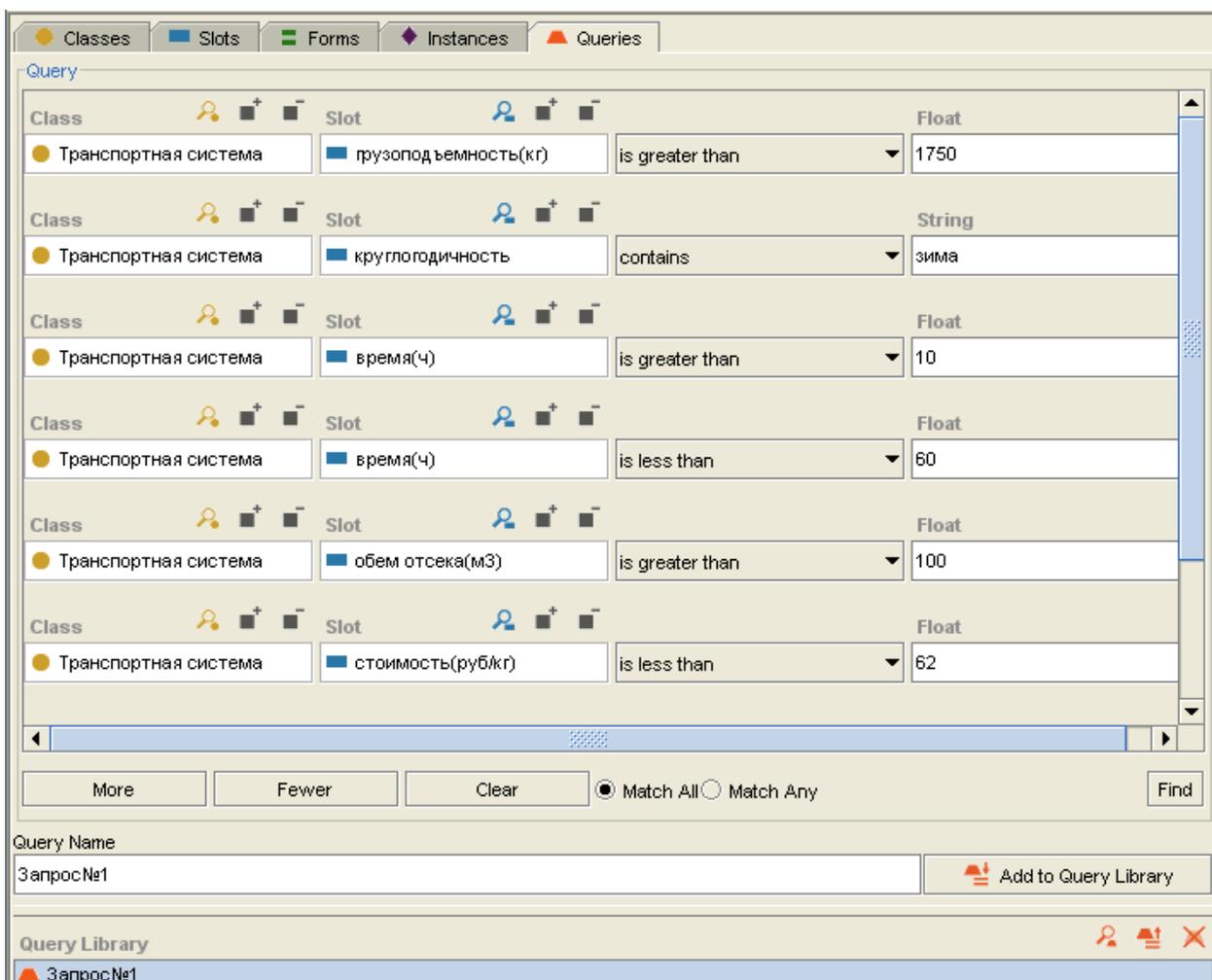


Рисунок 1.65 – Загрузка запроса

## 3.5.ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ

1. Дополнить созданную онтологию моделями самолетов в соответствии с номером варианта (Таблица 1.3).
2. Осуществить выбор транспортного средства для перевозки груза из одного населенного пункта в другой с учетом заданных параметров.
3. В качестве начального пункта перевозки груза рассмотреть город Москва, в качестве конечного – город Самара.
4. При осуществлении перевозки груза, учитывать возможность его разбиения на две, три или четыре части.

Таблица 1.3 – Варианты заданий

№ варианта	Включить онтологию следующие самолеты	Масса перевозимого груза (кг)	Объем перевозимого груза (м <sup>3</sup> )	Срок перевозки груза (ч)	Время года для осуществления перевозки	Максимальная стоимость перевозки (руб/кг)
1	АН-26Д	10 000	45	30-70	зима	50
	Lockheed C-5M SUPER GALAXY					
2	McDonnell Douglas C-17	1 000	100	10-30	лето	10
	C-80					
3	АН-124 РУСЛАН	80 000	150	90-100	осень	90
	Galaxy C-38					
4	Lockheed C-5M SUPER GALAXY	57 000	80	50-100	весна	70
	АН-26Д					
5	C-80	3 000	10	24	весна	66
	Boeing C-40					
6	АН-70	5 000	82	24-48	зима	15
	Lockheed C-5M SUPER GALAXY					
7	Boeing C-18	70 000	70	60-75	зима	82
	Lockheed C-130J SUPER HERCULES					
8	Shorts C-23 SHERPA	40 000	95	72	лето	53
	Boeing C-18					
9	ИЛ-76МФ	15 000	35	30	осень	34
	Lockheed C-130J SUPER HERCULES					
10	АН-124 РУСЛАН	1 500	8	40	весна	56
	Lockheed C-141C STARLIFTER					
11	Boeing C-32	25 000	100	80	зима	40
	Cessna U-27					
12	Cessna C-35	120 000	200	100-120	лето	67
	АН-124 РУСЛАН					
13	МиГ Т-101 ГРАЧ	11 000	10	56	лето	12
	Cessna UC-35					
14	АН-38Д	38 000	9	56-75	осень	15
	Boeing C-32					
15	Gulfstream C-37	60 000	20	80-100	зима	65
	АН-124					

## 1.2. Часть 2. Выбор самолета

### 3.1. Расширение онтологии

#### 1.2.1.1 Загрузка проекта

Для загрузки созданного проекта после запуска программы выберете его в окне приветствия по названию «*LR1\_varX*», где *X* номер Вашего варианта и нажмите кнопку «Open Recent». После этого откроется окно Вашего проекта в области просмотра классов (Classes) (Рисунок 1.66).

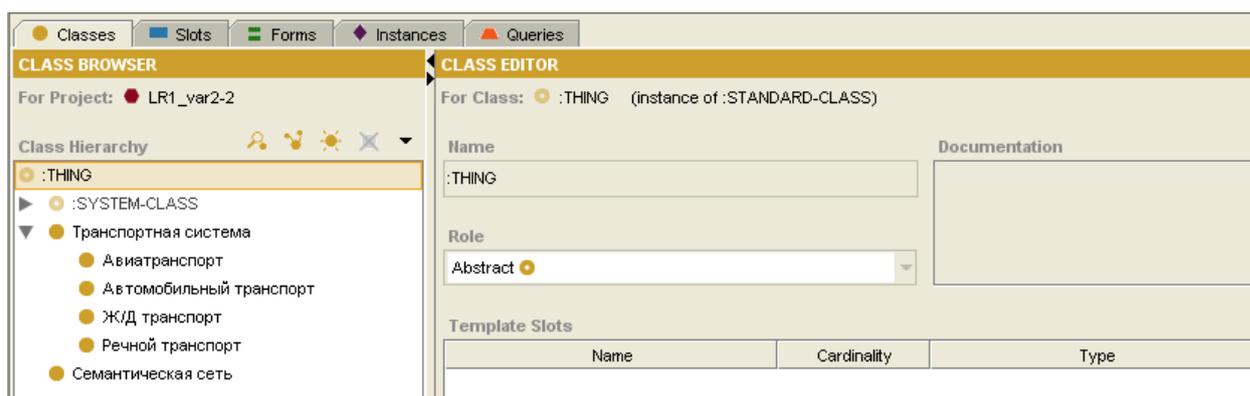


Рисунок 1.66 – Загрузка ранее созданного проекта

#### 1.2.1.2 Создание дополнительных слотов

Для того чтобы создать дополнительный слот, перейдите во вкладку Slots и нажмите кнопку  Create Slot. Измените стандартное имя в поле Name на «*взлетная масса(кг)*», в окне Value Type (тип) измените String на Float. В строке Minimum задайте значение «*0*» (Рисунок 1.67)

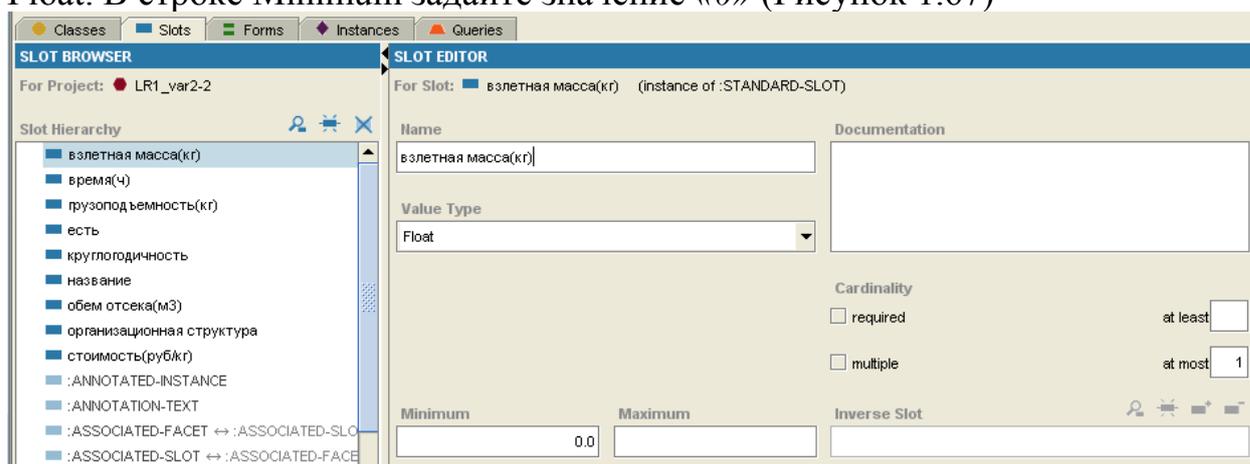


Рисунок 1.67 – Создание слота «взлетная масса»

Для включения созданного слота в класс в окне Domain нажмите кнопку  Add Class, из предложенного списка выберете класс «Авиатранспорт» и нажмите ОК (Рисунок 1.68).

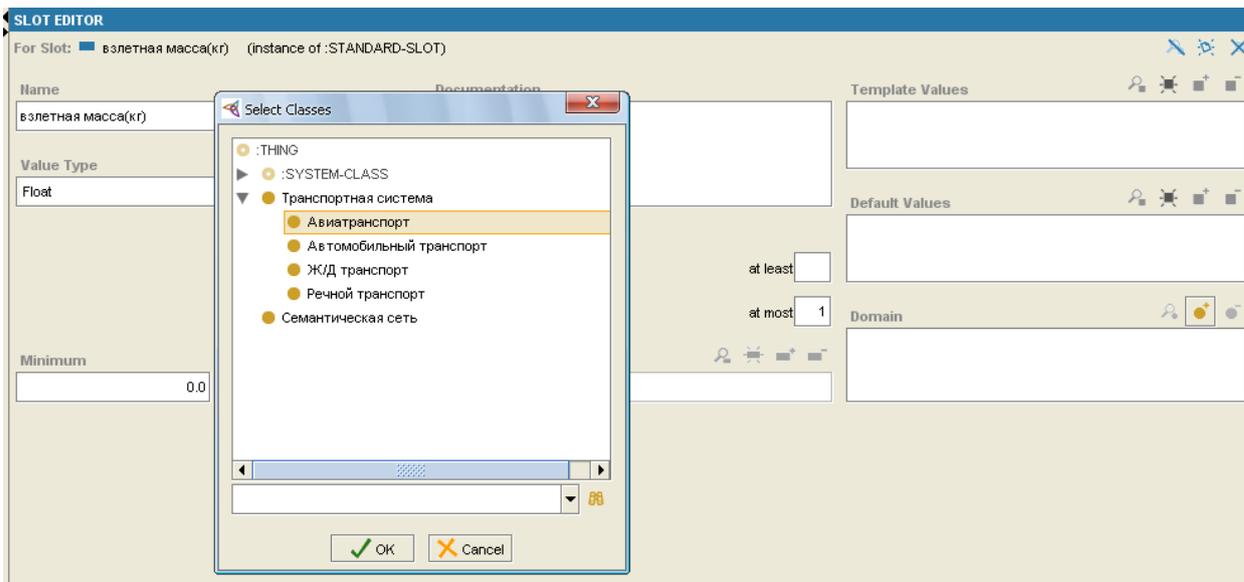


Рисунок 1.68 – Включение слота «взлетная масса» в класс «Авиатранспорт»

Аналогичным образом создайте слоты «дальность полета(км)» и «количество пассажиров», присвойте им тип Float, минимальное значение (Minimum) – «0» и включите их в класс «Авиатранспорт». Перейдите во вкладку Classes, выберете класс «Авиатранспорт», в окне Template Slots проверьте наличие новых слотов. Обратите внимание, что эти слоты включены только в класс «Авиатранспорт», поэтому знак ■ перед названием слота указан без скобок (Рисунок 1.69).

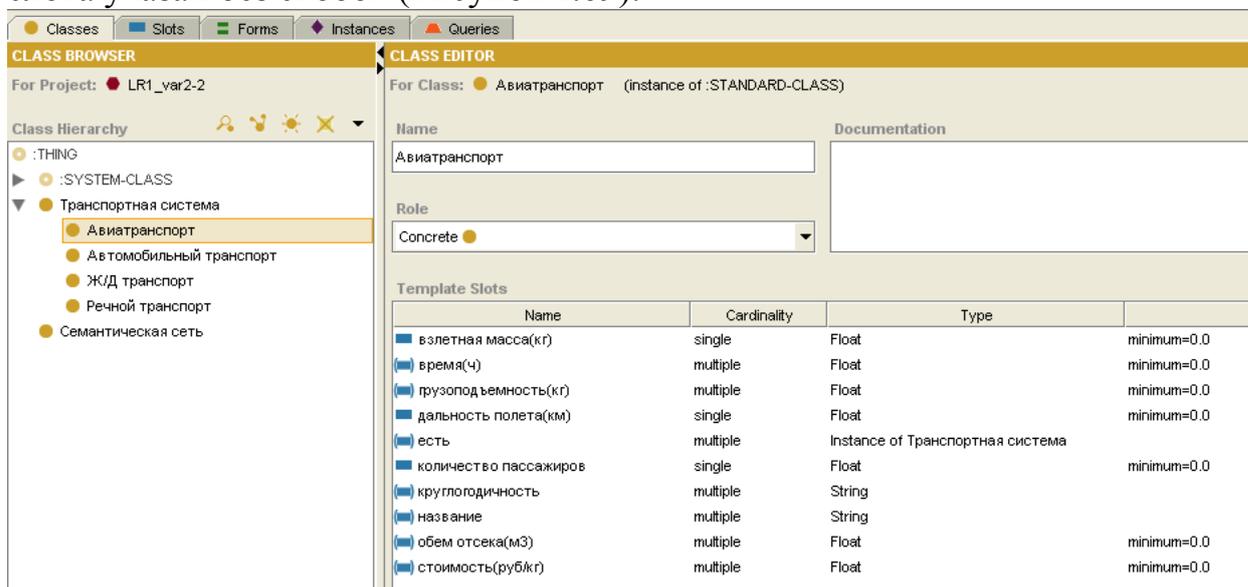


Рисунок 1.69 – Класс «Авиатранспорт»

### 1.2.1.3 Настройка форм виджетов созданных слотов

Перейдите в закладку форм (Forms) и выберете класс «Авиатранспорт». Вы увидите, что в нижней части окна появились новые слоты «взлетная масса», «дальность полета» и «количество пассажиров» в стандартной форме представления (Рисунок 1.70).

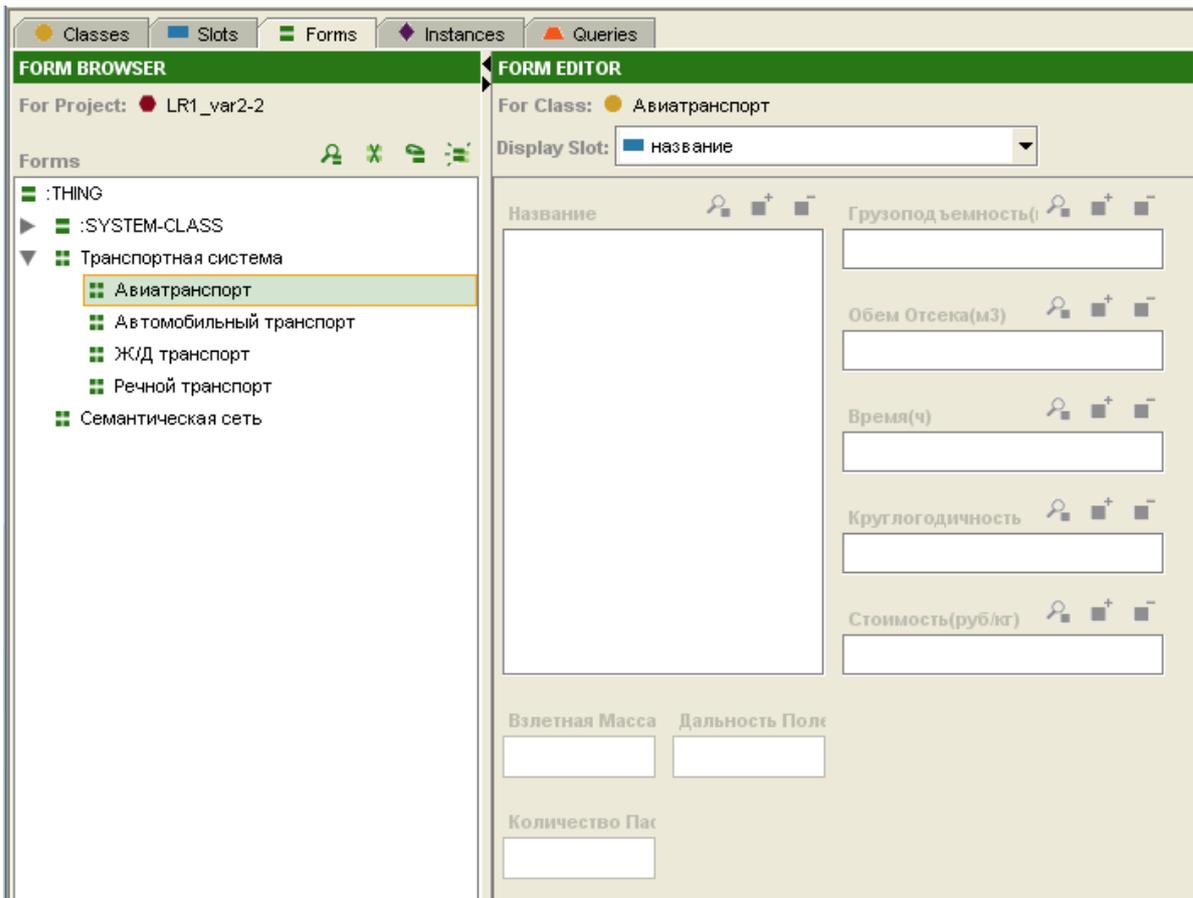


Рисунок 1.70 – Окно форм класса «Авиатранспорт»

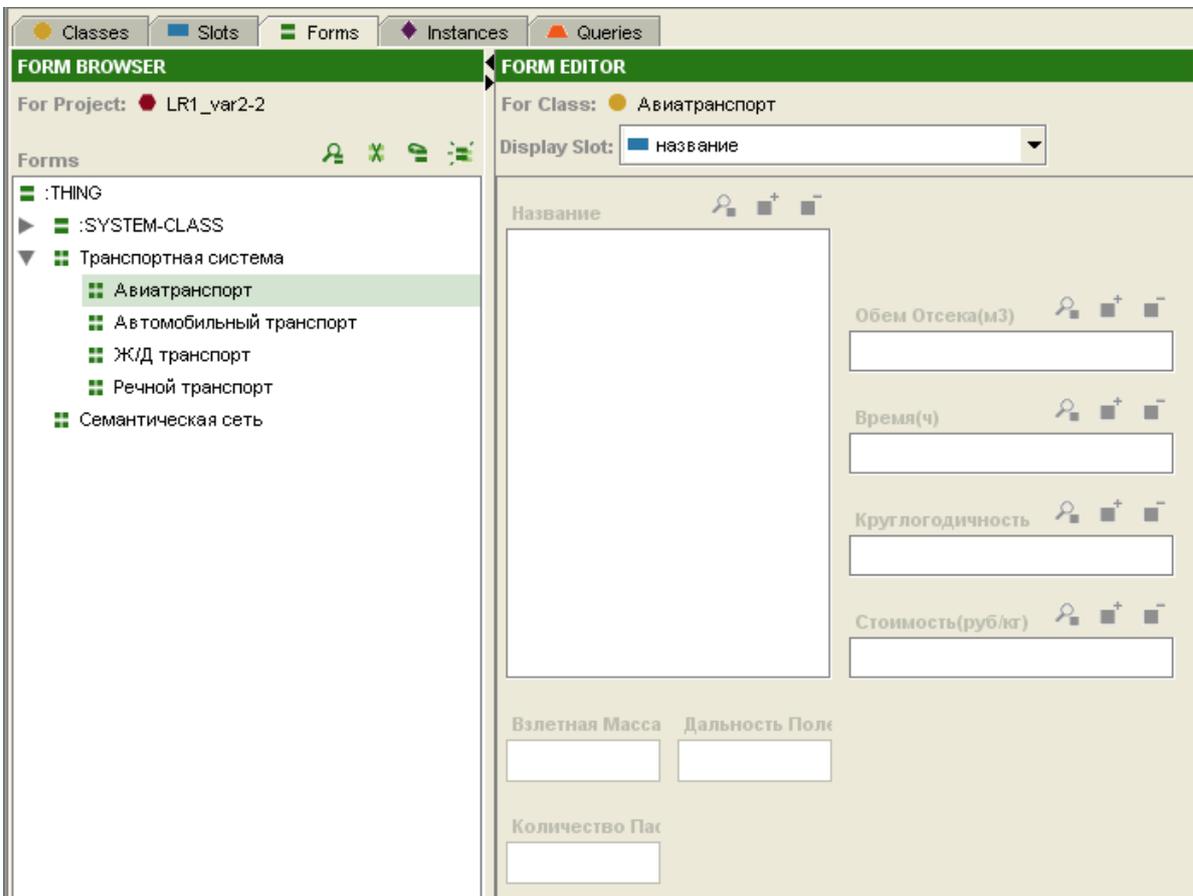


Рисунок 1.71 – Изменение видимости виджета «грузоподъемность»

Для этой части работы Вам понадобятся только виджеты слотов «название», «взлетная масса», «дальность полета» и «количество пассажиров», остальные необходимо скрыть.

Для того чтобы изменить видимость слотов в окне Form Editor кликнете на один из них (например «грузоподъемность») мышью и в правом верхнем углу в строке Selected Widget Type выберете <none> (Рисунок 1.71).

Аналогичным образом скройте виджеты слотов «объем отсека», «время», «круглогодичность» и «стоимость» (Рисунок 1.72).

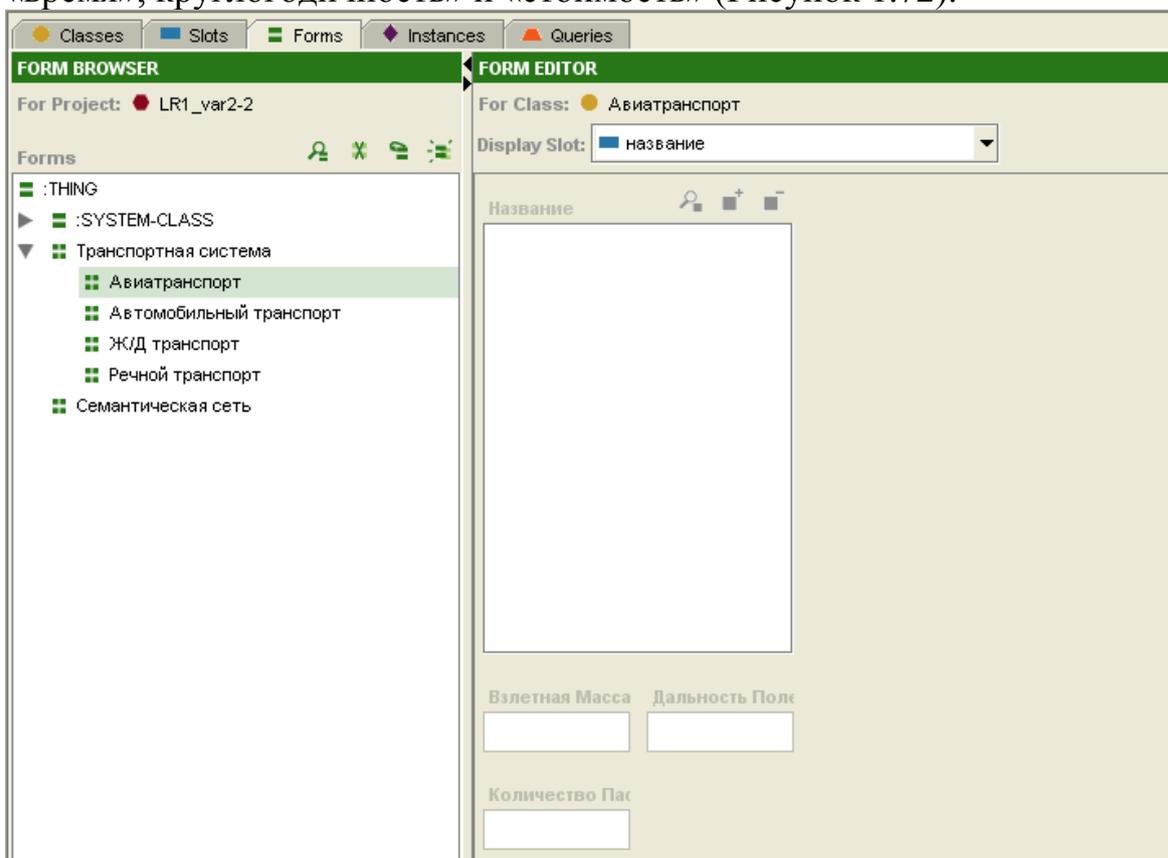


Рисунок 1.72 – Изменение видимости виджетов «объем отсека», «время», «круглогодичность», «стоимость»

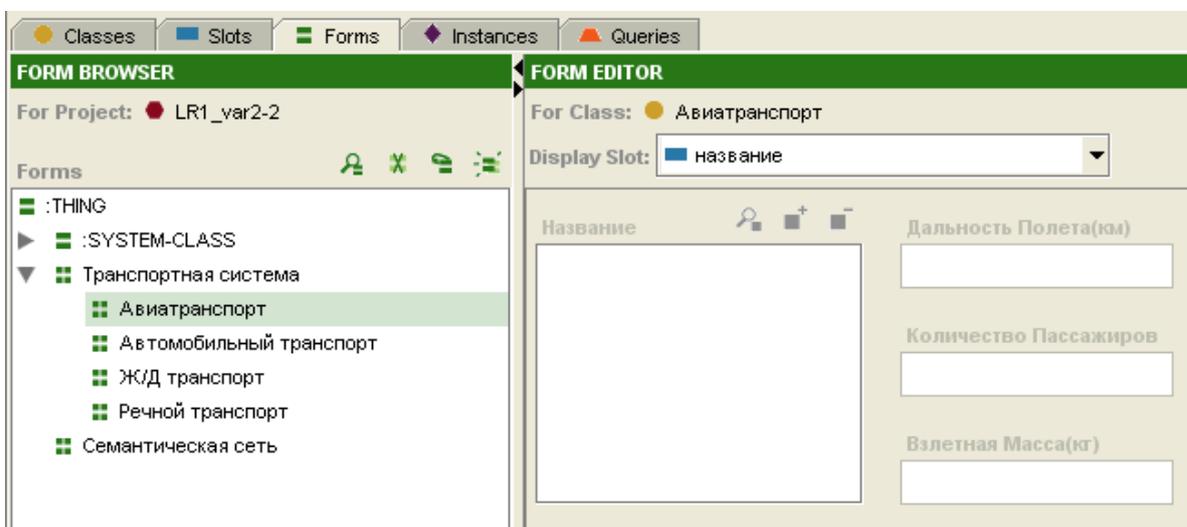


Рисунок 1.73 - Отредактированное окно Form Editor

Переместите виджеты «взлетная масса», «дальность полета» и «количество пассажиров» в соответствие с рисунком 1.73.

Подробнее о том, как изменять размер и местоположение виджета см. п.1.1.2.12 (Лабораторная работа №1. Часть1).

#### 1.2.1.4 Создание экземпляров классов

Перейдите на вкладку экземпляров (Instances) и выберете класс «Авиатранспорт». В окне Instances Browser Вы увидите грузовые самолеты созданные ранее (Рисунок 1.74).

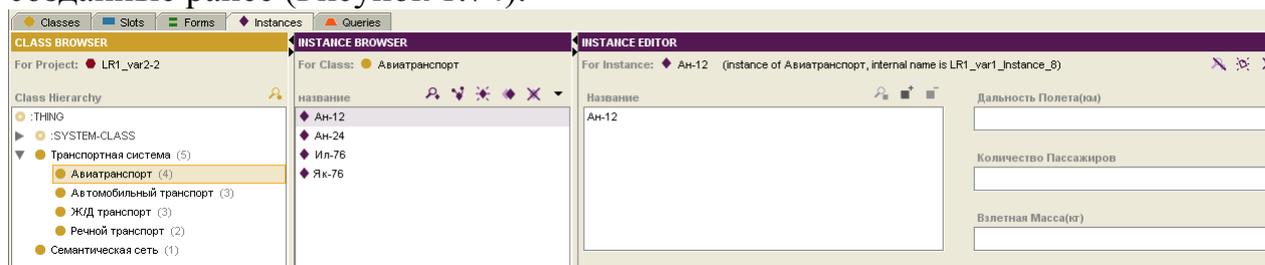


Рисунок 1.74 – Экземпляры класса «Авиатранспорт»

Для создания новых экземпляров класса «Авиатранспорт» в окне Instances Browser нажмите кнопку  Create Instances. В окне Instance Editor в строке «Название» нажмите кнопку  Add Value, в появившемся окне Create String Value введите с клавиатуры название самолета «Бритиш Аэро-спейс 146 - 300» и нажмите ОК. (Рисунок 1.75).

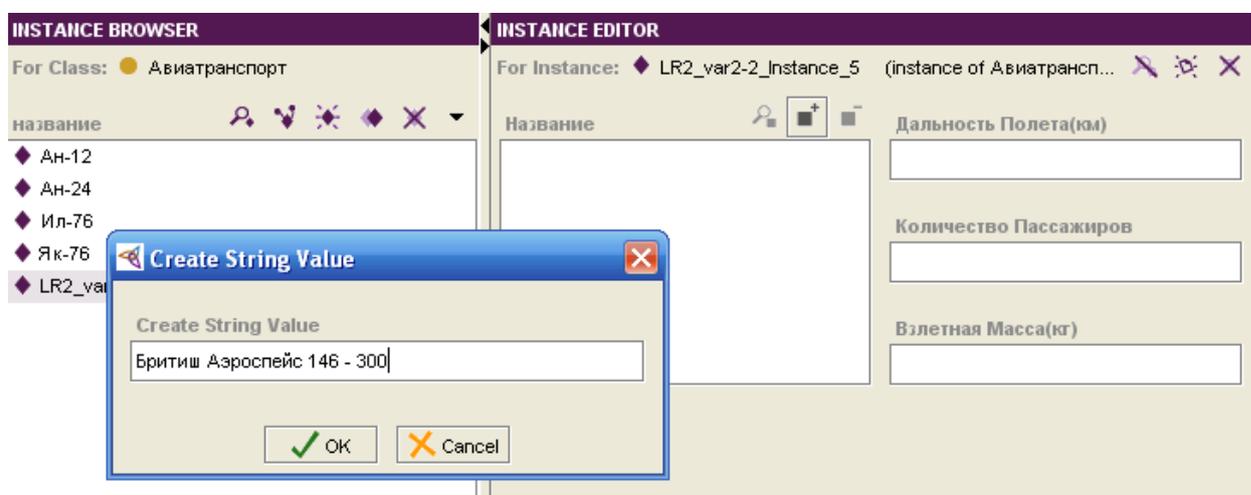


Рисунок 1.75 – Создание нового экземпляра

Аналогичным образом, нажимая на кнопку , в строках «взлетная масса», «дальность полета» и «количество пассажиров», введите данные (см. Таблица 1.4) для экземпляра «Бритиш Аэро-спейс 146 - 300» (Рисунок 1.76). Обратите внимание, что данные для этих слотов нужно вводить сразу в строку, т.к. в при их создании не была поставлена отметка в поле Multiple.

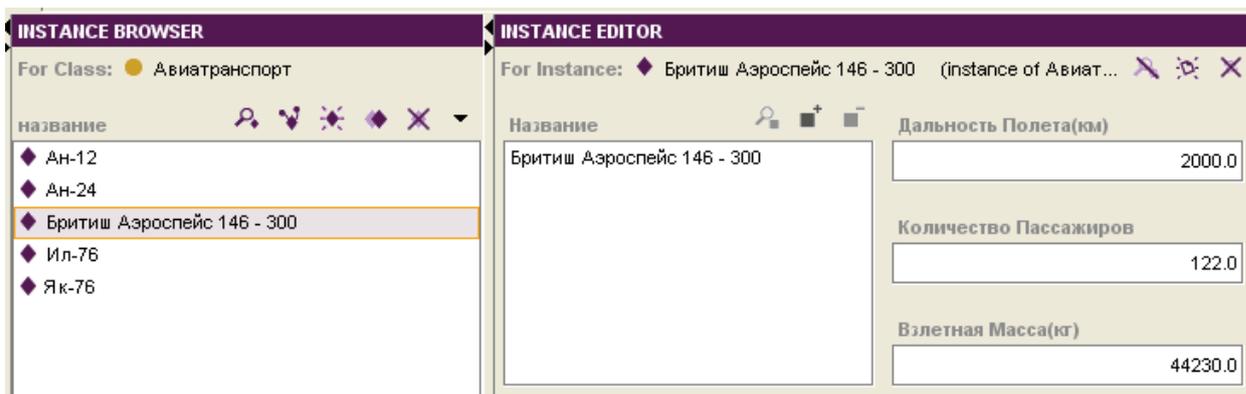


Рисунок 1.76 – Ввод значений для экземпляра «Бритиш Аэро-спейс 146 - 300»

Аналогичным образом создайте другие экземпляры, используя данные таблицы 1.4 (Рисунок 1.77).

Таблица 1.4 – Характеристики экземпляров

Тип самолета	Дальность полета, км	Количество пассажиров	Взлетная масса, кг
Бритиш Аэроспейс 146 - 300	2 000	122	44 230
Бритиш Аэроспейс 146 - 200	2 180	109	42 180
Бритиш Аэроспейс 146 - 100	1 730	93	38 010
Фоккер 70	2 000	79	36 740
Фоккер 100	2 390	107	43 090
Аэро Интернешнл Ридженд RJ100	2 260	100	46 000
ТУ - 134Б	2 020	96	47 600
ТУ – 334 - 100	2 000	110	46 100
ЯК – 42Д	2 150	120	56 500

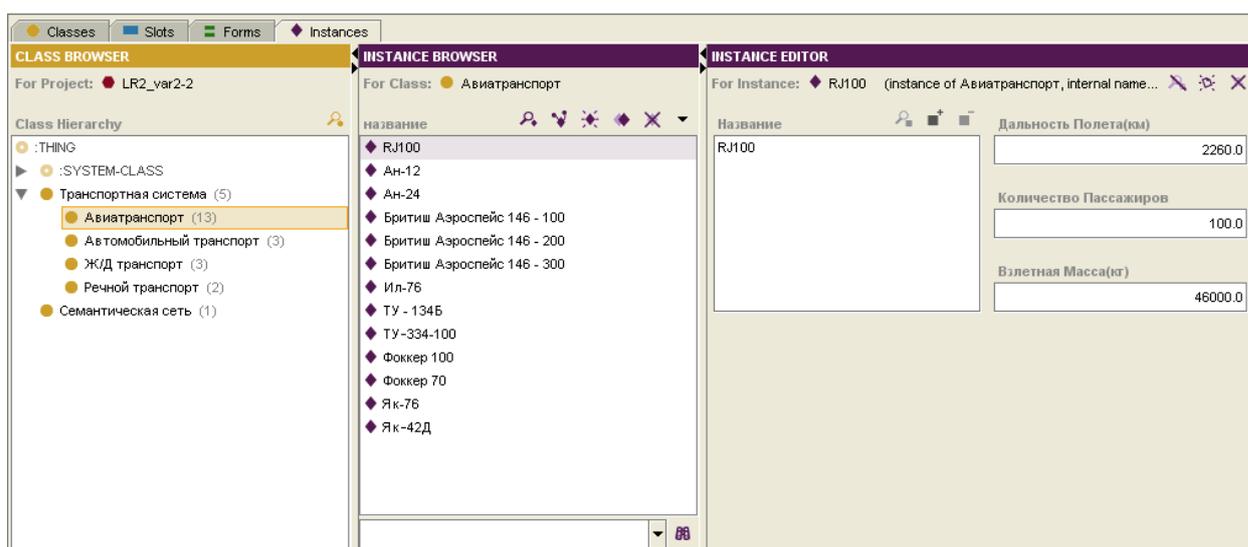


Рисунок 1.77 – Экземпляры класса авиатранспорт

### 1.2.1.5 Установка слотов отображения

В первой части лабораторной работы был выбран один слот отображения «название». Для выбора нескольких слотов отображения для

класса «Авиатранспорт», из всплывающего меню окна Instances Browser выберите Set Display Slot и нажмите на «Multiple Slots...» (Рисунок 1.78).

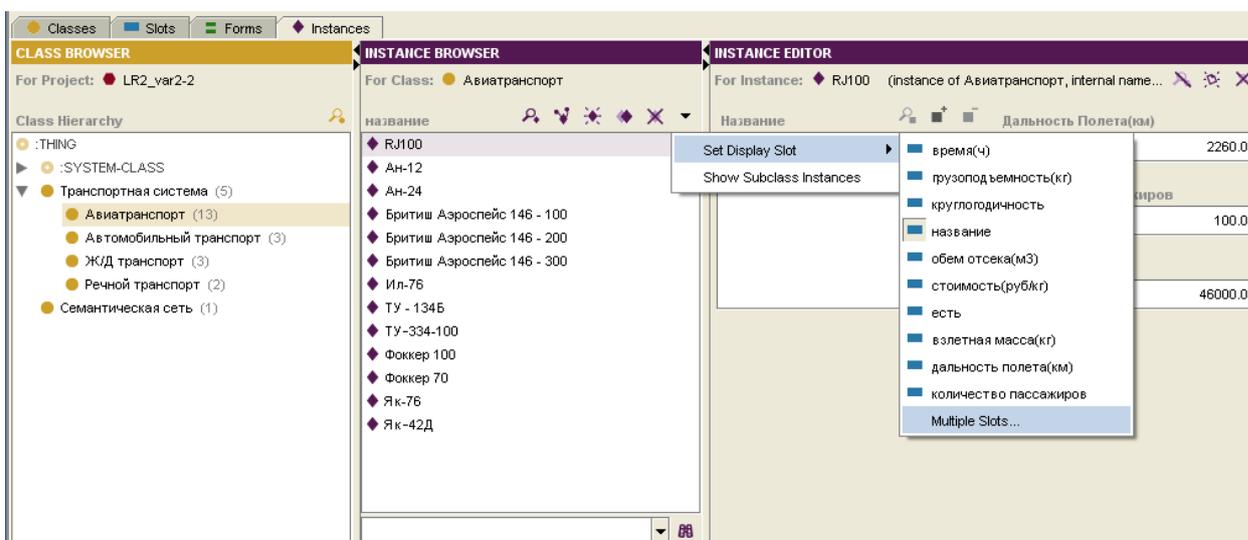


Рисунок 1.78 – Задание нескольких слотов отображения

После этого появится окно Multislot Display Pattern, в котором необходимо выбрать слоты отображения (Рисунок 1.79).

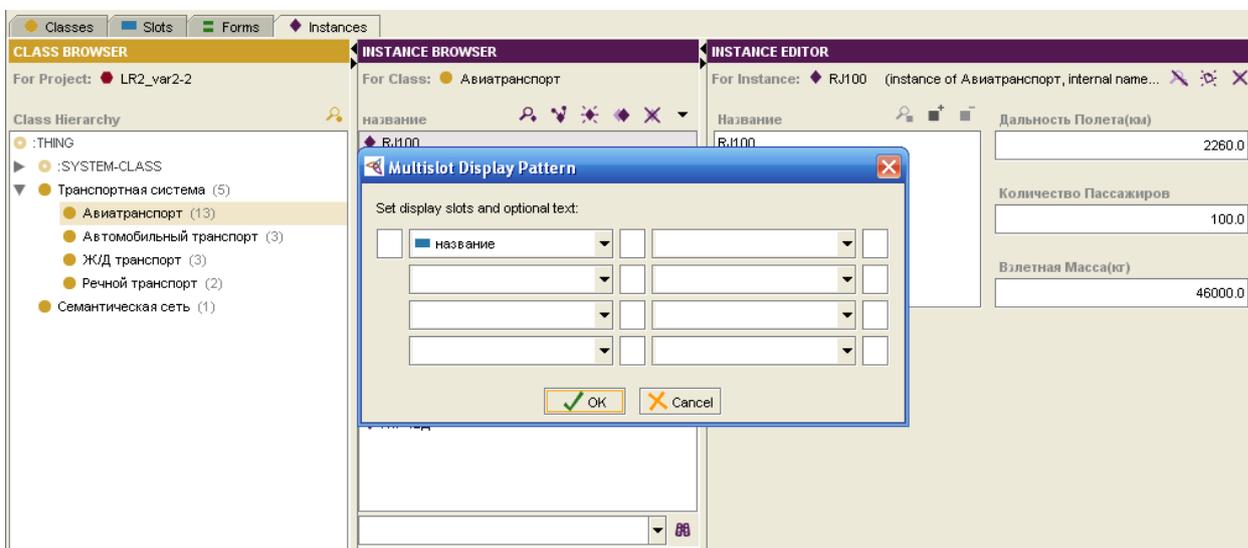


Рисунок 1.79 – Окно выбора слотов

Выберете из всплывающего меню слоты в порядке, показанном на рисунке 1.80. Нажмите ОК.

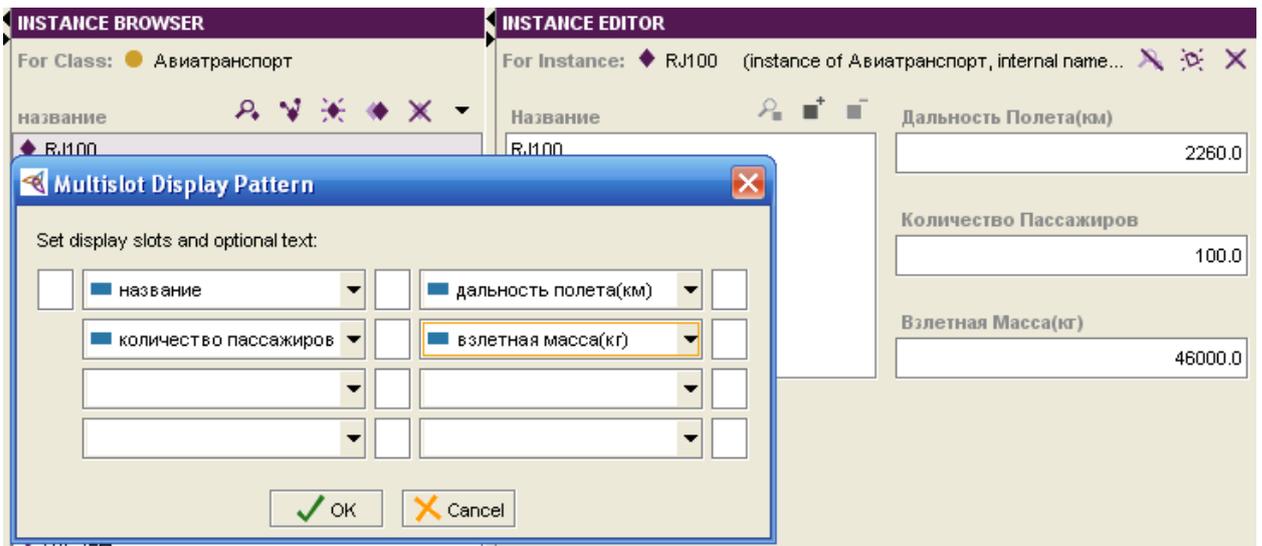


Рисунок 1.80 – Порядок выбора слотов

В окне Instances Browser появятся значения слотов, но без пробелов и названий (Рисунок 1.81).

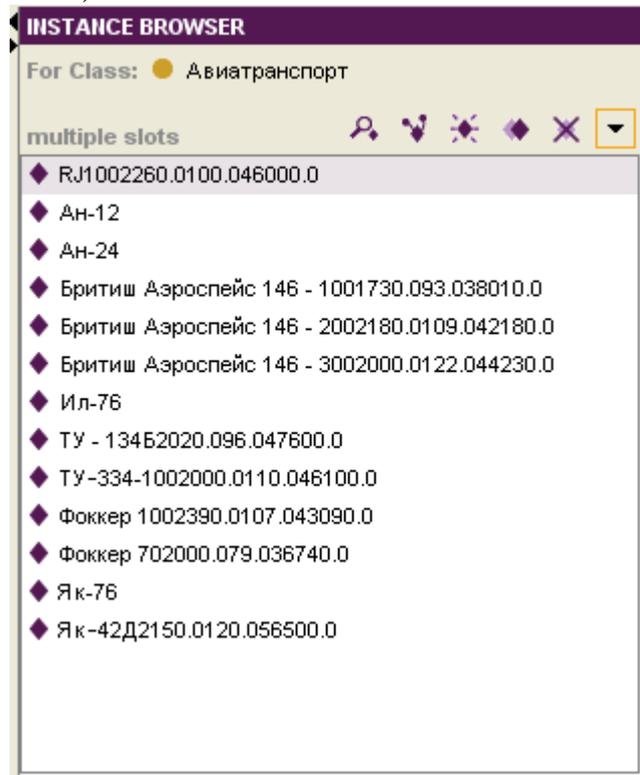


Рисунок 1.81 – Значения слотов экземпляра «Бритиш Аэро-спейс 146 - 100»

Откройте окно Multislot Display Pattern снова. Для этого из всплывающего меню окна Instances Browser выберите Set Display Slot и нажмите на «Multiple Slots...».

В окне Multislot Display Pattern находятся окна ввода комментариев. В работе они будут использованы для ввода обозначений дальности, количества и массы. Обратите внимание, что окно комментария находится слева от слота, а для второй и следующих строк (например «количество пассажиров»), на предыдущей. Введите следующие комментарии для слотов

(вместо знака подчеркивания используйте такое же число пробелов, кавычки ставить не нужно):

- «название» - «    »
- «дальность полета» - «    L    =    »,
- «количество пассажиров» - «    n    =    »,
- «взлетная масса» - «    m    =    ».

Сверьте введенные данные с рисунком 1.82. После этого нажмите ОК.

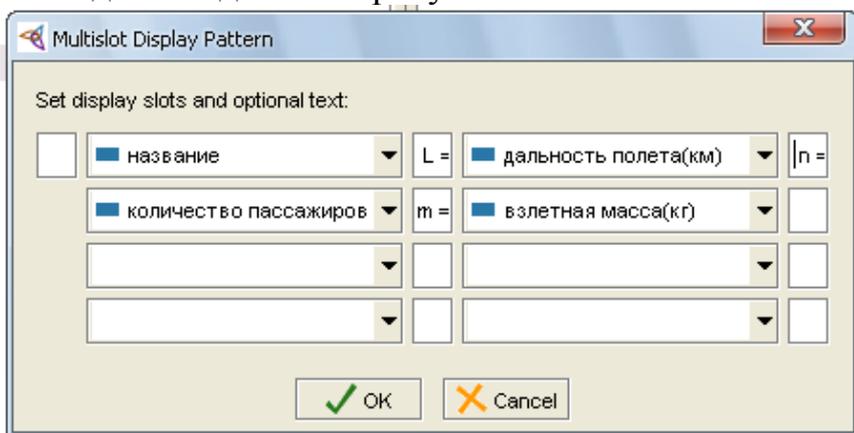


Рисунок 1.82 – Ввод комментариев для слотов

Отображение данных экземпляра изменится (Рисунок 1.83).

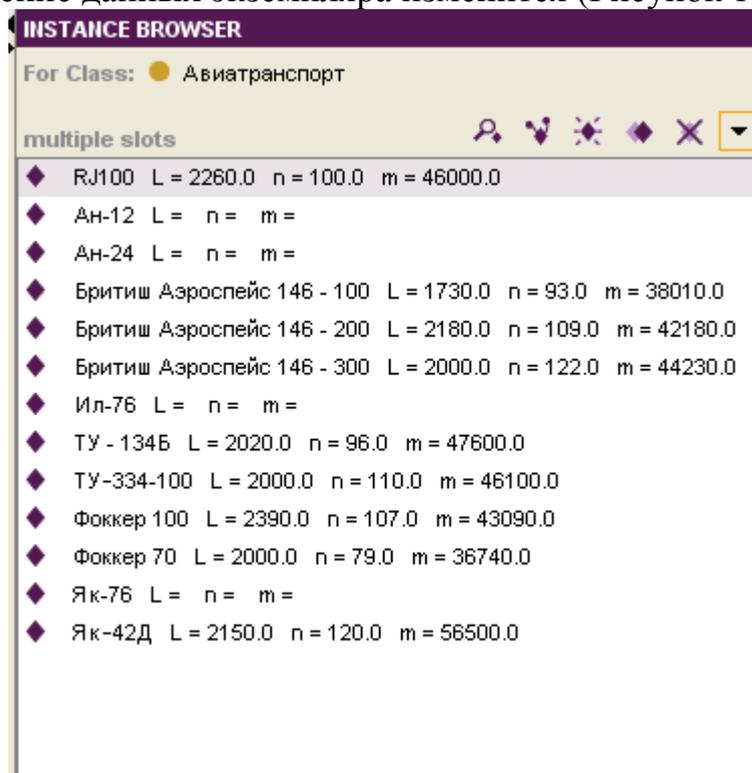


Рисунок 1.83 – Отредактированное окно экземпляров

### 1.2.1.6 Создание запроса

После создания онтологии необходимо осуществить выбор самолета по заданным параметрам с учетом их отклонений и минимизацией массы. Рассмотрим создание запроса на следующем примере (Таблица 1.5).

Таблица 1.5 – Данные для создания запроса

Дальность полета, км	Количество пассажиров	Отклонение, %	Взлетная масса, кг
2 000	100	+10	min

Для того чтобы создать запрос перейдите на закладку запросов (Queries). В строке Class нажмите кнопку  Select Class для выбора класса. В появившемся окне выберете класс «Авиатранспорт» и нажмите ОК (Рисунок 1.84).

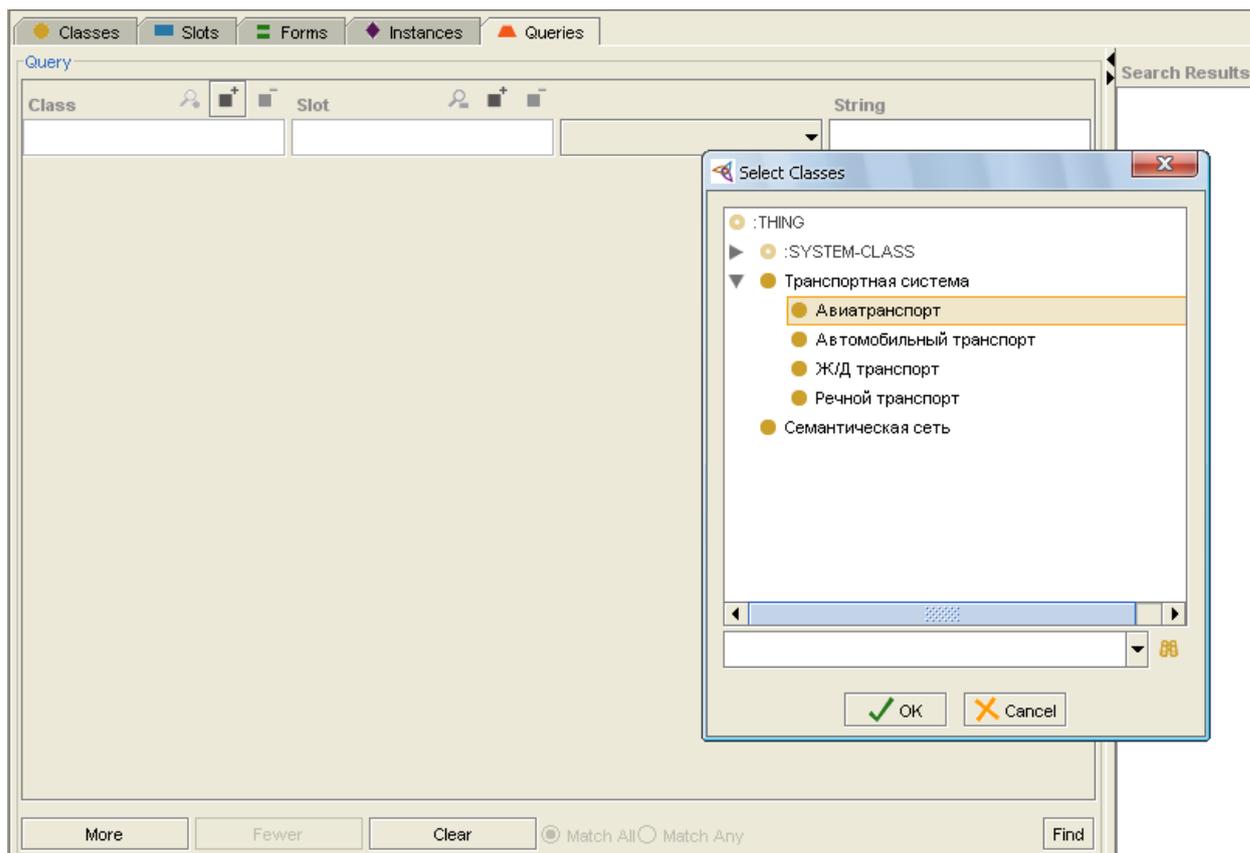


Рисунок 1.84 – Выбор класса для создания запроса

Для выбора слота в строке Slot нажмите кнопку  Select Slot. В появившемся окне (Рисунок 1.85) выберете «дальность полета» и нажмите ОК.

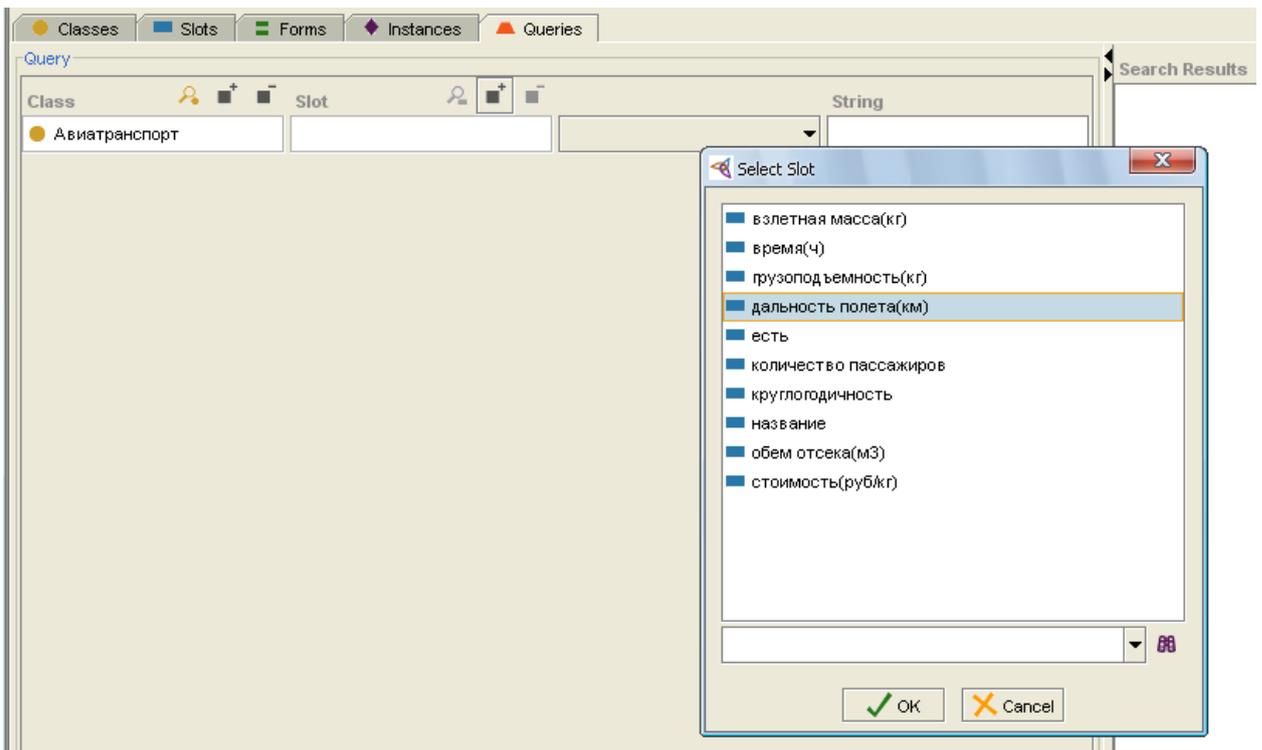


Рисунок 1.85 – Выбор слота для создания запроса

После выбора слота, меню справа от него стало активным и можно сделать выбор в соответствии с типом этого слота. В данном случае, тип Float – число с плавающей запятой (Рисунок 1.86).

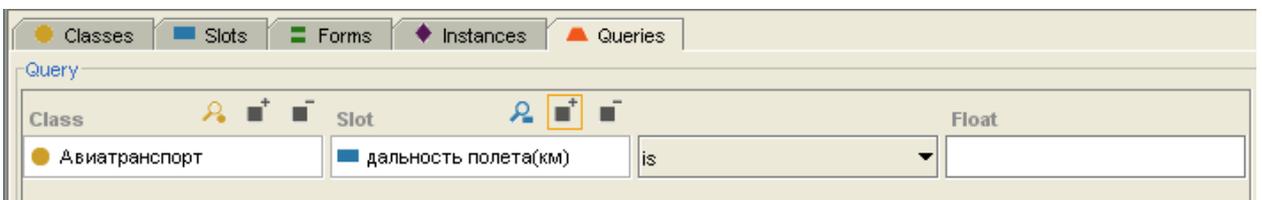


Рисунок 1.86 – Меню запроса

В выпадающем списке можно выбрать:

- is greater than – больше чем,
- is less than – меньше чем,
- is – равно.

Выберете «is greater than» (больше чем) и в следующую строку введите значение 2000 (Рисунок 1.87).



Рисунок 1.87 - Выбор параметров запроса

Для просмотра результата запроса нажмите кнопку «Find». В окне результатов запроса «Search Results» появится список транспортных средств с дальностью полета более 2 000 (Рисунок 1.88).



Рисунок 1.88 – Результат запроса по дальности полета

При создании запроса необходимо учитывать отклонение  $\pm 10\%$ , то есть удовлетворять условиям запроса будут самолеты, имеющие дальность полета от 1800 км до 2200 км. Измените в соответствии с этими условиями запрос (Рисунок 1.89). Для добавления новой строки запроса используйте кнопку More.

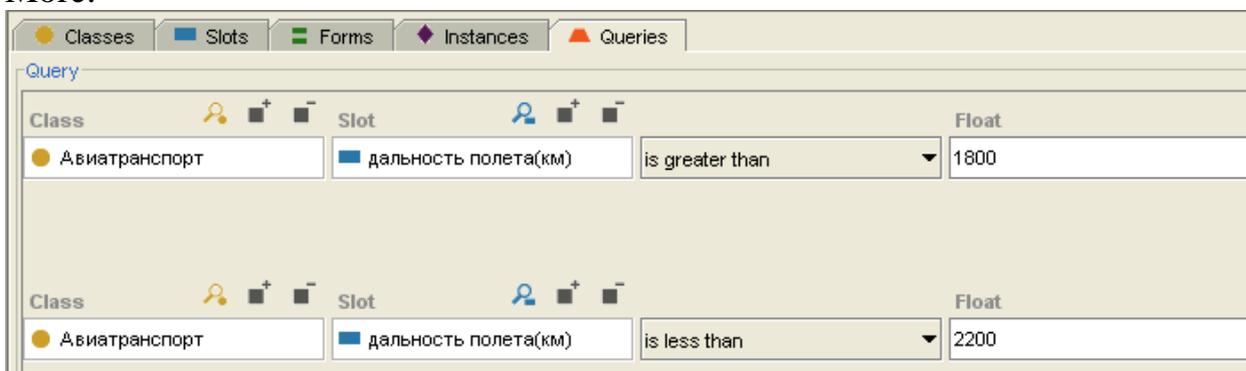


Рисунок 1.89 – Редактирование запроса с учетом отклонений

Аналогичным образом создайте еще две строки запроса для добавления ограничений по количеству пассажиров с учетом отклонений (Рисунок 1.90).

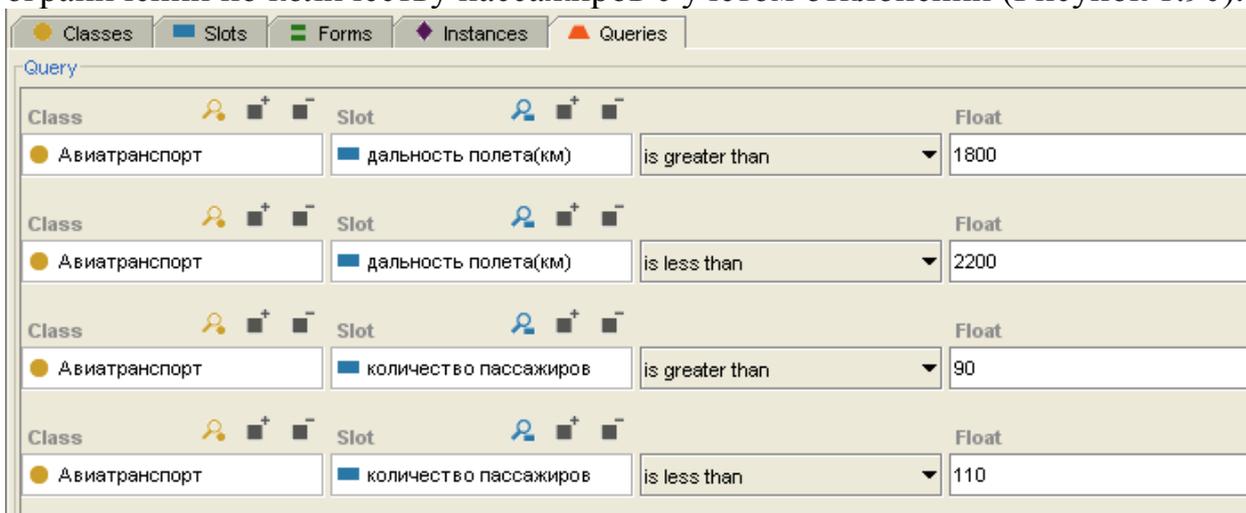


Рисунок 1.90 – Запрос с учетом отклонений

Для просмотра результата запроса нажмите кнопку «Find». В окне результатов запроса «Search Results» появится список транспортных средств удовлетворяющих условиям (Рисунок 1.91).

Search Results (2)	
◆	Бритиш Аэроспейс 146 - 200 L = 2180.0 n = 109.0 m = 42180.0 (Авиатранспорт)
◆	Ту - 134Б L = 2020.0 n = 96.0 m = 47600.0 (Авиатранспорт)

Рисунок 1.91 – Результат запроса

Условиям запроса удовлетворяют два самолета Бритиш Аэроспейс 146-200 и Ту 134Б. По условиям задания необходимо выбрать самолет с минимальной взлетной массой. Этим условиям удовлетворяет самолет Бритиш Аэроспейс 146-200, имеющий массу  $m=42\ 180$  кг, дальность полета  $L=2\ 180$  км и перевозящий количество пассажиров  $n=109$ .

### 3.2.ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ

1. Включить в онтологию самолеты в соответствии с полученным вариантом задания (Таблица 1.6).
2. Осуществить выбор самолета, удовлетворяющий заданным условия по дальности полета и числу пассажиров с учетом заданных отклонений.
3. Окончательный выбор осуществить по минимальной массе

Таблица 1.6 – Варианты заданий

№ варианта	Включить в онтологию следующие самолеты	Дальность полета, км	Количество пассажиров	Отклонение, %
1	АН-26Б	2 000	98	±3
	Aero Spacelines GUPPY-201			
2	АН-70	1 500	100	±10
	Aero Spacelines B-377 PREGNANT GUPPY			
3	АН-124 РУСЛАН	2 500	150	±15
	Boeing 747-400			
4	АН-225 МРИЯ	2 300	200	±10
	Miles M.69 MARATHON II			
5	АН-124-100 РУСЛАН	2 000	150	±8
	McDonnell Douglas MD-11F			
6	ИЛ-114Т	1 900	150	±10
	Let L-410 TURBOLET			
7	ТУ-330	2 200	120	±5
	Aero Spacelines B-377			
8	Boeing 747-400	2 000	120	±5
	АН-124 РУСЛАН			
9	Boeing 757-200	2 500	130	±8
	ИЛ-114Т			
10	Boeing 767-300	2 600	100	±3
	АН-70			
11	Airbus A3XX-F	2 400	150	±10
	АН-124 РУСЛАН			

12	McDonnell Douglas MD-11F	1 900	130	±15
	Miles M.69 MARATHON II			
13	Антонов АН-8	1 800	90	±12
	Aero Spacelines B-377 PREGNANT GUPPY			
14	Breguet Br.761 DEUX PONTS	1 700	105	±9
	ИЛ-114Т			
15	АН-26Б	2 300	130	±5
	Boeing 747-400			

### 3.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое онтология?
2. Что такое база знаний?
3. Цель создания онтологии?
4. Какие этапы включает в себя процесс построения онтологии в Protege?
5. Как определить, правильно ли создана онтология?
6. На каком этапе создания онтологии нельзя изменить иерархию классов?
7. Сколько слотов может включать в себя класс?
8. Может ли слот быть связан с несколькими классами? Каким образом?
9. С помощью какой команды можно произвести наследование слотов базового класса для его подкласса?
10. Можно ли создавать отношения между классами не определенными в иерархии классов? Каким образом?
11. Сколько экземпляров может иметь класс?
12. Что такое слот отображения? Для чего он используется?
13. Какое количество слотов отображения может быть задано?
14. Какой параметр указывает на то, что виджет был изменен?
15. Для чего нужна библиотека запросов?

## ЛАБОРАТОРНАЯ РАБОТА №2

### ВВЕДЕНИЕ

На основании нижеприведенных характеристик самолетов, информация о которых хранится в базе данных, необходимо рассчитать и выбрать удельную нагрузку на крыло (часть 1) и потребную тяговооруженность самолета (часть 2)

Основные характеристики, а также требования к проектируемому самолету могут быть описаны в виде *онтологии*. В качестве редактора онтологии используется продукт компании MAGENTA CORPORATION LIMITED. Проектируемый самолет следует рассматривать как *проект* или *заказ*, а объект, необходимый для выполнения вычислений – как *ресурс*. На основании результатов процесса поиска взаимного соответствия между заказом и ресурсами (*матчинга*) принимаются или пересматриваются решения о бронировании или освобождении ресурсов (т.е., устанавливаются связи между заказом и адекватными ему ресурсами). Тем самым выполняются необходимые расчеты значений атрибутов концептов. Для этого использовались упрощенные зависимости, построенные на основе формул [4] и статистики по магистральным самолетам.

В качестве исходных данных для выполнения работы требуются следующие характеристики самолетов:

для первой части:

- эффективная механизация, да/нет;
- удельная нагрузка на крыло, даН/м<sup>2</sup>;
- длина разбега по ВПП, м;
- количество двигателей на самолете, шт.
- максимальное аэродинамическое качество,
- число Маха;
- высота полета, м.

для второй части:

- дальность полета, км;
- крейсерская высота полета, м;
- число Маха;
- число пассажиров, чел.;
- число пассажиров в ряду, чел.;
- эффективная механизация, да/нет.

**Цель лабораторной работы № 2** – освоение метода онтологического анализа проектных процедур по выбору параметров самолета путем разработки приложений, использующих скрипты для расчетов в процессе одностороннего матчинга при принятии решений.

В процессе выполнения лабораторной работы № 2 решаются следующие **задачи**:

- освоение инструментов, предоставляемых конструктором онтологий и исполняющей системой;
- освоение приемов проектирования дескриптивной онтологии и онтологии мира заказов/ресурсов;
- освоение приемов конструирования скриптов для расчета значений атрибутов концептов;
- освоение приемов конструирования и моделирования онтологической сцены;
- изучение структуры агентов заказа/ресурса.

В начале каждого раздела излагаются общие теоретические положения и приводятся принятые упрощенные зависимости по определению параметров проектируемого самолета.

## 1.1. Часть 1. Выбор удельной нагрузки на крыло

### 2.1.1. Расчет условия обеспечения заданной скорости захода на посадку

Условие обеспечения заданной скорости захода на посадку рассчитывается по формуле (2.1)

$$p'_0 \leq \frac{C_{y_{\max \text{ нос}}} \cdot V_{з.п.}^2}{30,2 \cdot (1 - \overline{m_T})}, \quad (2.1)$$

где:

- $p'_0$  - удельная нагрузка на крыло при заходе на посадку,  $даН/м^2$ ;
- $C_{y_{\max \text{ нос}}}$  - значение максимального коэффициента подъемной силы самолета в посадочной конфигурации; значение  $C_{y_{\max \text{ нос}}}$  выбирается на основе статистических данных в зависимости от системы механизации крыла. Для упрощения решения задачи принято:
  - для эффективной механизации  $C_{y_{\max \text{ нос}}} = 2.5$ ;
  - для слабой (неэффективной) механизации  $C_{y_{\max \text{ нос}}} = 2.0$ .
- $V_{з.п.}$  - скорость захода на посадку, м/с; значение  $V_{з.п.}$  выбирается на основе статистических данных. В данной работе принято значение:  $V_{з.п.} = 230 \text{ км/ч} = 63,89 \text{ м/с}$ ;
- $\overline{m_T}$  - предполагаемое значение относительной массы топлива.

С учетом выбранных значений статистических данных удельная нагрузка на крыло при заходе на посадку рассчитывается по следующим упрощенным формулам (2.2) и (2.3) в зависимости от выбранной эффективной механизации (исходное данное)

- для эффективной механизации  $p'_0 = 10204.83 / (30.2(1 - \overline{m_T}))$ ; (2.2)

- для неэффективной механизации  $p'_0 = 8163.86 / (30.2(1 - \overline{m_T}))$ . (2.3)

Предполагаемое значение относительной массы топлива рассчитывается по формуле (2.4)

$$\overline{m_T} = 0.075 + \frac{0.0175 \cdot L_p}{V_{кр}}, \quad (2.4)$$

где:

- $L_p$  – расчетная дальность полета, км (исходное данное);
- $V_{кр}$  - крейсерская скорость полёта, м/с.

Крейсерская скорость полета рассчитывается по формуле (2.5)

$$V_{кр} = M * 20.04679 * \sqrt{288.15 - 41318.979 * H_{кр} / (6356766 + H_{кр})}, \quad (2.5)$$

где:

- $M$  - число Маха (исходное данное);
- $H_{кр}$  - крейсерская высота полёта, м (исходное данное).

### 2.1.2. Расчет условия обеспечения заданной крейсерской скорости на расчетной высоте полета

Условие обеспечения заданной крейсерской скорости на расчетной высоте полета определяется по формуле (2.6)

$$P_0'' \leq \frac{0.208 * \Delta H_{кр} * V_{кр}^2 * \sqrt{C_{x_0}}}{1 - 0.6 * \overline{m}_T}, \quad (2.6)$$

где:

- $P_0''$  - удельная нагрузка на крыло при заданной крейсерской скорости на расчетной высоте полета,  $daH/m^2$  ;
- $V_{кр}$  - крейсерская скорость полёта, м/с, рассчитывается по формуле (2.5);
- $\overline{m}_T$  - предполагаемое значение относительной массы топлива, рассчитывается по формуле (2.4);
- $\Delta H_{кр}$  - относительная плотность воздуха на расчетной высоте, м;
- $C_{x_0}$  - коэффициент лобового сопротивления при нулевой подъемной силе.

Относительная плотность воздуха на расчетной высоте определяется по формуле (2.7)

$$\Delta H_{кр} = 0.8 - 0.000046 * H_{кр}, \quad (2.7)$$

где:  $H_{кр}$  - крейсерская высота полёта, м (исходное данное).

Коэффициент лобового сопротивления при нулевой подъемной силе рассчитывается по формуле (2.8)

$$C_{x_0} = (0,882 + 0,147M) \cdot \left[ 0,015288 + 0,000656 \cdot \lambda_\phi + \frac{0,041}{\lambda_\phi^2} \right]. \quad (2.8)$$

где:

- $M$  - число Маха (исходное данное);
- $\lambda_\phi$  - удлинение фюзеляжа.

Удлинение фюзеляжа рассчитывается по формуле (2.9)

$$\lambda_{\phi} = \frac{0.906 * n_{vr} + 1.323 * n_{pas}}{(0.649 * n_{vr} + 0.6) * n_{vr}}. \quad (2.9)$$

где:

- $n_{vr}$  - число пассажиров в ряду, чел. (исходное данное выбираемое из диапазона 3 – 10 чел.);
- $n_{pas}$  - число пассажиров, чел. (исходное данное из технического задания).

Максимальное аэродинамическое качество приближенно рассчитывается по формуле (2.10)

$$K_{\max} = \frac{1}{0,43\sqrt{C_{x0}}} + 1. \quad (2.10)$$

где:

- $K_{\max}$  – максимальное аэродинамическое качество;
- $C_{x0}$  - коэффициент лобового сопротивления при нулевой подъемной силе, рассчитывается по формуле (8).

### 2.1.3. Расчет удельной нагрузки на крыло

Расчетное значение удельной нагрузки на крыло определяется по формуле (2.11)

$$P_0 = \min (p'_0, p''_0). \quad (2.11)$$

где:

- $P_0$  – расчетное значение удельной нагрузки на крыло,  $\text{даН/м}^2$  ;
- $p'_0$  - удельная нагрузка на крыло при заходе на посадку,  $\text{даН/м}^2$ ; рассчитывается по формуле (2.1);
- $p''_0$  - удельная нагрузка на крыло при заданной крейсерской скорости на расчетной высоте полета,  $\text{даН/м}^2$  ; рассчитывается по формуле (2.6).

## 2.1.4. Проектирование дескриптивной онтологии

### 2.1.4.1 Создание онтологии

Загрузите конструктор онтологий (файл *OntCons.exe*). Создайте новую библиотеку онтологий (*File* → *New*). По умолчанию она имеет имя *OntologyLibrary\_1*. Переименуйте ее, вводя в поле *Name* имя *OntologyLibrary\_Wing Load*. Создайте дескриптивную онтологию предметной области «Выбор удельной нагрузки на крыло» (*New Item* → *Descriptive ontology*). По умолчанию дескриптивная онтология имеет имя *Ontology\_1*. Переименуйте ее, вводя в поле *Name* имя *Ontology\_Wing Load*.

### 2.1.4.2 Создание и удаление концепта

После создания онтологии при нажатии на кнопку <+> можно открыть дерево, узлами которого являются категории концептов онтологии. Это абстрактные базовые классы, и от них необходимо наследовать классы-потомки для построения собственной онтологии. Создание потомка осуществляется посредством выбора концепта, который будет являться предком создаваемого концепта, нажатия правой кнопки мыши и выбора пункта меню *New Item*. Созданный концепт можно удалить, выделив его и нажав клавишу <Del>, либо выбрав в контекстном меню пункт *Delete*.

#### – Концепт «объект»

Концепт «объект» - это сущность, которая присутствует в мире, описанном в онтологии. После создания дескриптивной онтологии ПО «Выбор удельной нагрузки на крыло» необходимо создать два концепта «объект».

Для этого создайте объект *Wing\_Demand* (*Objects* → *New Item* → *Object*), переименуйте его в *Wing\_Demand*, изменяя значение в поле *Name*, назначьте этому объекту пиктограммы, соответствующие трем видам отображения. Аналогичным образом создайте объект *Wing\_Resource*, переименуйте его в *Wing\_Resource* и назначьте этому объекту пиктограммы.

Каждый концепт «объект» может иметь определенный список атрибутов.

#### – Концепт «атрибут»

Концепт «атрибут» – это величина, характеризующая объект (количественное выражение признака).

– Создайте **булевский атрибут** (*Attributes* → *New Item* → *Boolean Attribute*), переименуйте созданный атрибут в *Effective\_mehan*.

– Создайте **целые атрибуты** (*Attributes* → *New Item* → *Integer Attribute*), переименуйте созданные атрибуты в:

*\_H\_kr, \_L, \_n\_pas, \_n\_vr, X, Y*.

– Создайте **вещественные атрибуты** (*Attributes* ->*New Item* -> *Float Attribute*), переименуйте созданные атрибуты в:

*\_M, Motn\_topl, V\_kr, DeltaH\_kr, K\_max, CX\_0, Lambda\_fuz, P0\_zahpos, P0\_H, P0.*

### **2.1.4.3 Создание связей между концептами**

Создание связей между концептами осуществляется посредством механизма *Drag&Drop*. Например, для добавления атрибута к списку атрибутов объекта необходимо «перетащить» нужный атрибут на концепт объекта.

Для того, чтобы установить связи между созданными концептами дескриптивной онтологии ПО «Выбор удельной нагрузки на крыло», т.е. указать, что объект *Wing\_Demand* имеет атрибуты *\_Effective\_mehan, \_H\_kr, \_M, \_L, \_n\_pas, \_n\_vr, Motn\_topl, V\_kr, DeltaH\_kr, CX\_0, Lambda\_fuz, P0\_zahpos, P0\_H, P0, K\_max, X, Y*, а объект *Wing\_Resource* – атрибуты *X, Y*, необходимо в дереве концептов «перетащить» атрибуты *\_Effective\_mehan, \_H\_kr, \_M, \_L, \_n\_pas, \_n\_vr, Motn\_topl, V\_kr, DeltaH\_kr, CX\_0, Lambda\_fuz, P0\_zahpos, P0\_H, P0, K\_max, X, Y* на объект *Wing\_Demand*. Аналогично следует «перетащить» атрибуты *X, Y* на объект *Wing\_Resource*.

Просмотреть список атрибутов данного объекта можно в редакторе свойств концепта «объект».

В результате, у объекта в закладке *Uses* появится список имен атрибутов, а у атрибута в закладке *Used by* появится имя объекта (объектов), использующего этот атрибут. На рисунке 2.1 показаны свойства объекта *Wing\_Demand* в закладке *Uses*, причем данный объект имеет атрибуты *\_Effective\_mehan, \_H\_kr, \_M, \_L, \_n\_pas, \_n\_vr, Motn\_topl, V\_kr, DeltaH\_kr, CX\_0, Lambda\_fuz, P0\_zahpos, P0\_H, P0, K\_max, X, Y* (соответствующие связи отображены в закладке *Uses*). В закладке *Used by* атрибутов отображена их связь с соответствующим объектом.

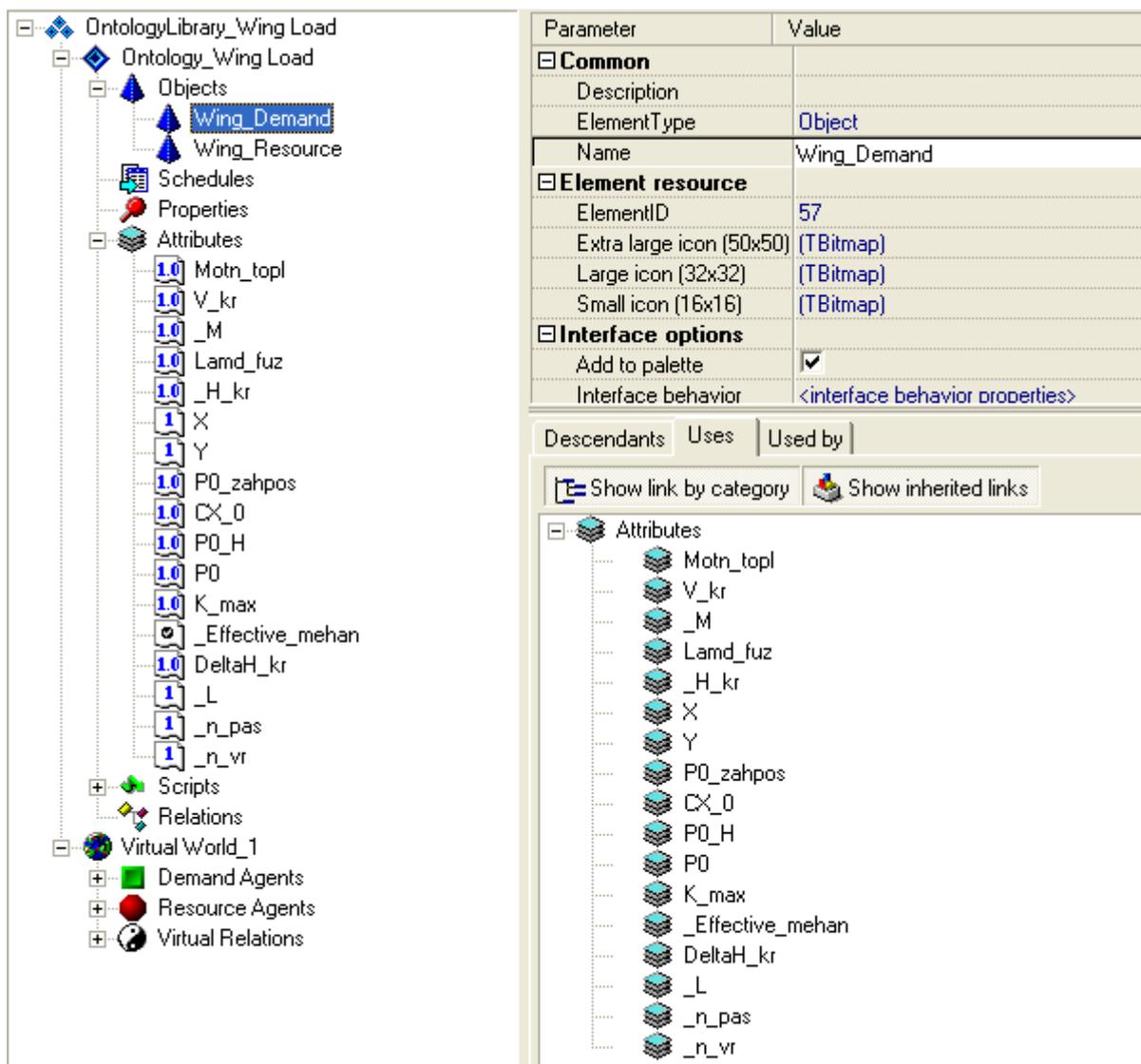


Рисунок 2.1 - Атрибуты объекта *Wing\_Demand* в закладке *Uses*

### – Концепт «скрипт»

Концепт «скрипт» – это определенное правило расчета некоторого значения, записанное на языке программирования. В конструкторе онтологий для написания скриптов используется подмножество языка Object Pascal. Скрипты следует использовать, если необходимо рассчитать значение атрибута в зависимости от значений других атрибутов. В скрипте можно использовать только те концепты и их параметры (например, атрибуты для объекта и т.п.), которые являются параметрами скрипта (т.е., находятся в закладке *Uses* скрипта).

Для того чтобы указать, что некоторый концепт является параметром скрипта, необходимо «перетащить» этот концепт на соответствующий концепт «скрипт». В свою очередь, результат вычислений, производимых в скрипте, должен быть связан с каким-либо атрибутом соответствующего объекта. Для этого необходимо концепт «скрипт» перетащить на нужный объект.

Далее следует вызвать редактор скриптов и написать, либо отредактировать тело скрипта (вызвать *Other* -> *Script*  в редакторе свойств скрипта, либо нажать кнопку  в закладке *Script body*).

#### – Определение скрипта для вычисления крейсерской скорости полета

Чтобы вычислить с помощью скрипта крейсерскую скорость полета по формуле (2.5), необходимо создать концепт «скрипт», который будет вычислять крейсерскую скорость полета. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item* -> *Script*. Созданный скрипт переименуйте в *V\_kr Calculate* и свяжите его с атрибутом *V\_kr*, т.е. перетащите скрипт на атрибут *V\_kr*. Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *V\_kr Calculate*, вычисляющего крейсерскую скорость полета, являются атрибуты *\_M* и *H\_kr* (рис. 2.2). Напишите тело скрипта. Для этого выделите скрипт *V\_kr Calculate* в дереве концептов дескриптивной онтологии, затем перейдите в закладку *Script body*, нажмите на кнопку . При этом откроется окно редактора скриптов, в котором необходимо набрать следующий текст (имена концептов заключаются в кавычки, незначащие пробелы в начале идентификаторов не допускаются). Имена концептов необходимо выбирать в списке концептов, которые являются параметрами скрипта. Скрипт заканчивается точкой с запятой.

**begin**

```
Result := "_M" * 20.04679 * sqrt( 288.15 - 41318.979 * "_H_kr"/  
(6356766+"_H_kr") );
```

**end;**

Сохраните скрипт нажатием на кнопку . Закройте окно редактора скриптов. Проверьте правильность синтаксиса скрипта нажатием на кнопку *Check syntax*. Если будет зафиксирована ошибка, необходимо вызвать редактор скриптов и внести необходимые исправления.

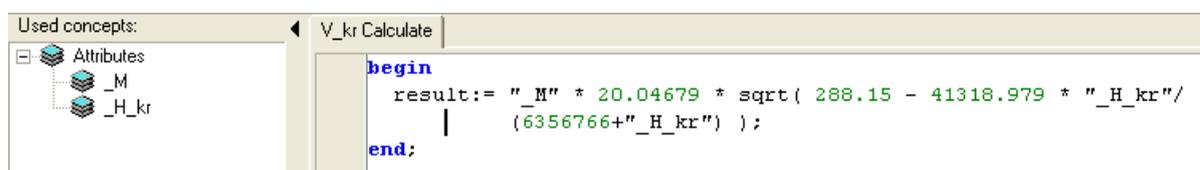


Рисунок 2.2 – Атрибуты и тело скрипта *V\_kr Calculate*

Аналогичным образом создайте следующие скрипты:

- **Определение скрипта для вычисления относительной массы топлива** (по формуле 2.4).

Созданный скрипт переименуйте в *Motn\_topl Calculate* и свяжите его с атрибутом *Motn\_topl*. Параметрами скрипта *Motn\_topl Calculate* являются атрибуты *\_L* и *V\_kr*. Напишите тело скрипта (рис. 2.3).

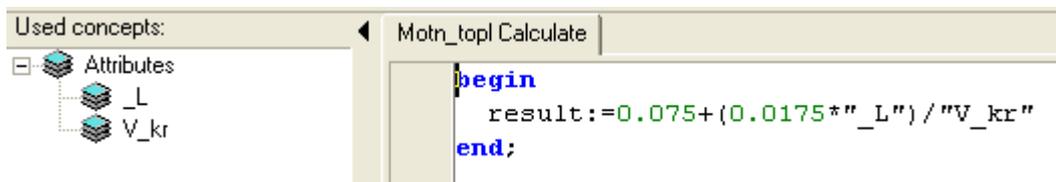


Рисунок 2.3 – Атрибуты и тело скрипта *Motn\_topl Calculate*

- **Определение скрипта для вычисления удельной нагрузки на крыло при заходе на посадку** (по формулам 2.2 и 2.3).

Созданный скрипт переименуйте в *P0\_zahpos Calculate* и свяжите его с атрибутом *P0\_zahpos*. Параметрами скрипта *P0\_zahpos Calculate* являются атрибуты *\_Effective\_mehan* и *Motn\_topl*. Напишите тело скрипта (рис. 2.4).

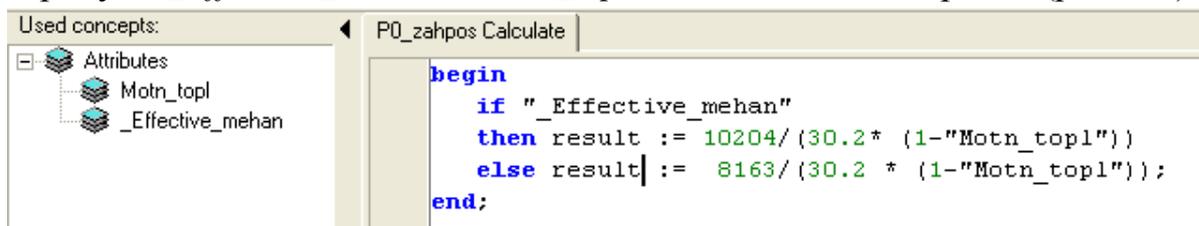


Рисунок 2.4 – Атрибуты и тело скрипта *P0\_zahpos Calculate*

- **Определение скрипта для вычисления относительной плотности воздуха на расчетной высоте** (по формуле 2.7).

Созданный скрипт переименуйте в *DeltaH\_kr Calculate* и свяжите его с атрибутом *DeltaH\_kr*. Параметрами скрипта *DeltaH\_kr Calculate* является атрибут *\_H\_kr*. Напишите тело скрипта (рис. 2.5).

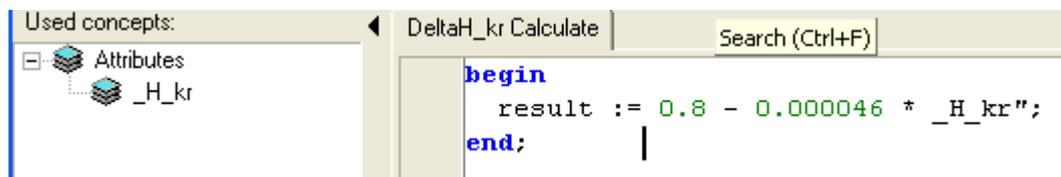


Рисунок 2.5 – Атрибуты и тело скрипта *DeltaH\_kr Calculate*

- **Определение скрипта для вычисления удлинения фюзеляжа** (по формуле 2.9).

Созданный скрипт переименуйте в *Lambda\_fuz Calculate* и свяжите его с атрибутом *Lambda\_fuz*. Параметрами скрипта *Lambda\_fuz Calculate*,

вычисляющего удлинение фюзеляжа, являются атрибуты *\_n\_pas* и *\_n\_vr*. Напишите тело скрипта (рис. 2.6).

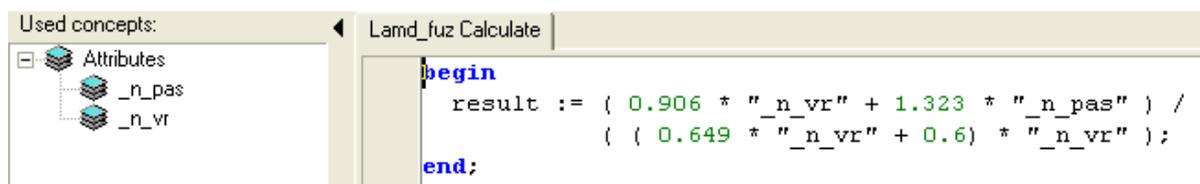


Рисунок 2.6 – Атрибуты и тело скрипта *Lambda\_fuz Calculate*

– **Определение скрипта для вычисления коэффициента лобового сопротивления при нулевой подъемной силе (по формуле 2.8).**

Созданный скрипт переименуйте в *CX\_0 Calculate* и свяжите его с атрибутом *CX\_0*. Параметрами скрипта *CX\_0 Calculate*, вычисляющего коэффициент лобового сопротивления при нулевой подъемной силе, являются атрибуты *\_M* и *Lamd\_fuz*. Напишите тело скрипта (рис. 2.7).

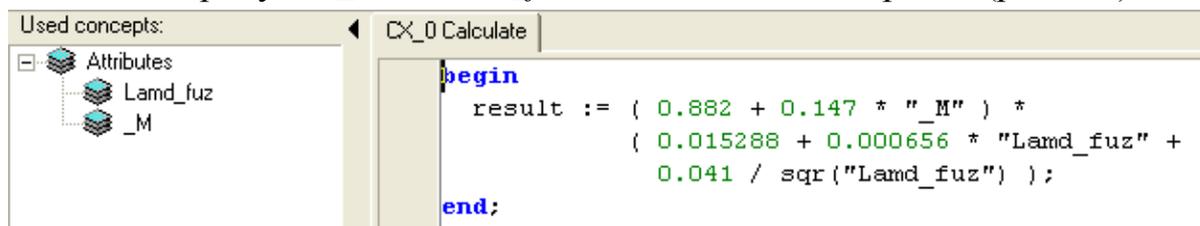


Рисунок 2.7 – Атрибуты и тело скрипта *CX\_0 Calculate*

– **Определение скрипта для вычисления удельной нагрузки на крыло при заданной крейсерской скорости на расчетной высоте полета (по формуле 2.6).**

Созданный скрипт переименуйте в *PO\_H Calculate* и свяжите его с атрибутом *PO\_H*. Параметрами скрипта *PO\_H Calculate*, вычисляющего коэффициент лобового сопротивления при нулевой подъемной силе, являются атрибуты *V\_kr*, *DeltaH\_kr*, *CX\_0* и *Motn\_topl*. Напишите тело скрипта (рис. 2.8).

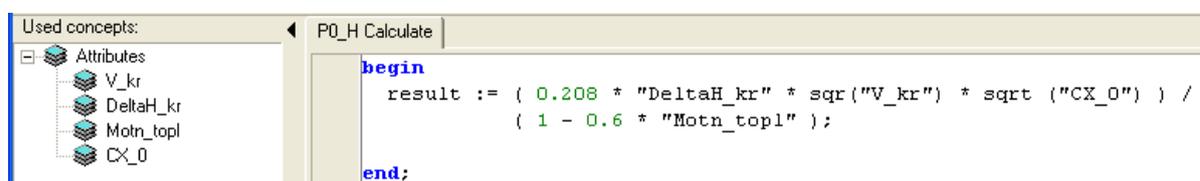


Рисунок 2.8 – Атрибуты и тело скрипта *PO\_H Calculate*

– **Определение скрипта для вычисления удельной нагрузки на крыло (по формуле 2.11).**

Созданный скрипт переименуйте в *PO Calculate* и свяжите его с атрибутом *PO*. Параметрами скрипта *PO Calculate*, вычисляющего

коэффициент лобового сопротивления при нулевой подъемной силе, являются атрибуты *PO\_zahpos* и *PO\_H*. Напишите тело скрипта (рис. 2.9).

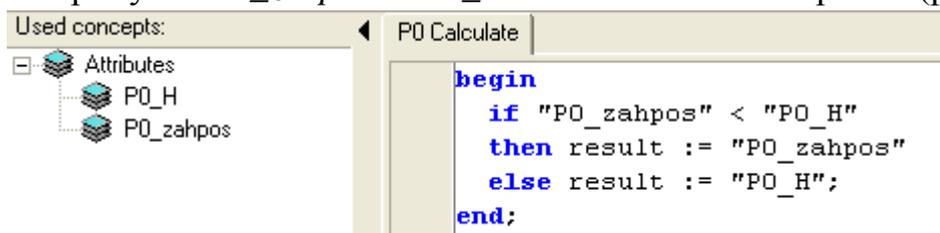


Рисунок 2.9 – Атрибуты и тело скрипта *PO Calculate*

– **Определение скрипта для вычисления максимального аэродинамического качества (по формуле 2.10).**

Созданный скрипт переименуйте в *K\_max Calculate* и свяжите его с атрибутом *K\_max*. Параметрами скрипта *K\_max Calculate*, вычисляющего максимальное аэродинамическое качество, является атрибут *CX\_0*. Напишите тело скрипта (рис. 2.10).

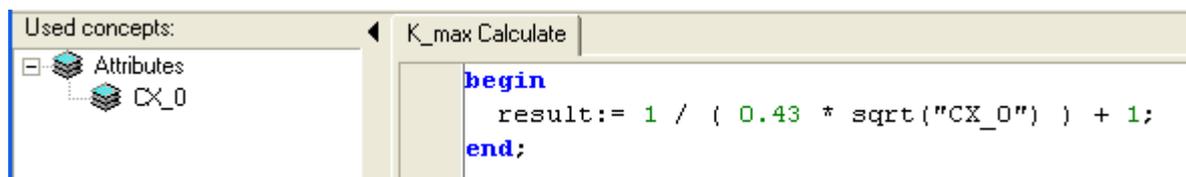


Рисунок 2.10 – Атрибуты и тело скрипта *K\_max Calculate*

Таким образом, в дереве концептов дескриптивной онтологии предметной области “Выбор удельной нагрузки на крыло” можно видеть следующие скрипты (рис. 2.11).

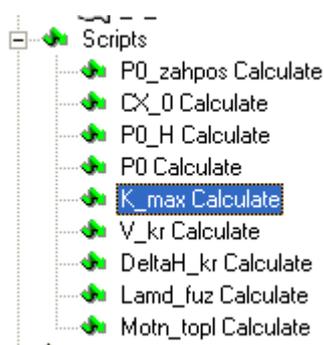


Рисунок 2.11 – Скрипты дескриптивной онтологии

#### 2.1.4.4 Представление онтологии в виде семантической сети

Дескриптивная онтология может быть представлена не только в виде дерева концептов, но также в виде семантической сети, представляющей собой ориентированный граф, в котором вершины представляют собой концепты онтологии, а ребра отображают связи между концептами.

Пользователь имеет возможность перемещать концепты семантической сети в пределах экрана, перетаскивая их с помощью мыши.

Для того чтобы получить представление дескриптивной онтологии в виде семантической сети, необходимо выполнить следующую последовательность команд *Tools -> Ontology as network*.

В левой части открывшегося окна *Ontology Network* дескриптивная онтология отображается в виде дерева концептов, а в правой части – в виде семантической сети (рис. 2.12).

При закрытии окна *Ontology Network* происходит возврат в конструктор онтологий.

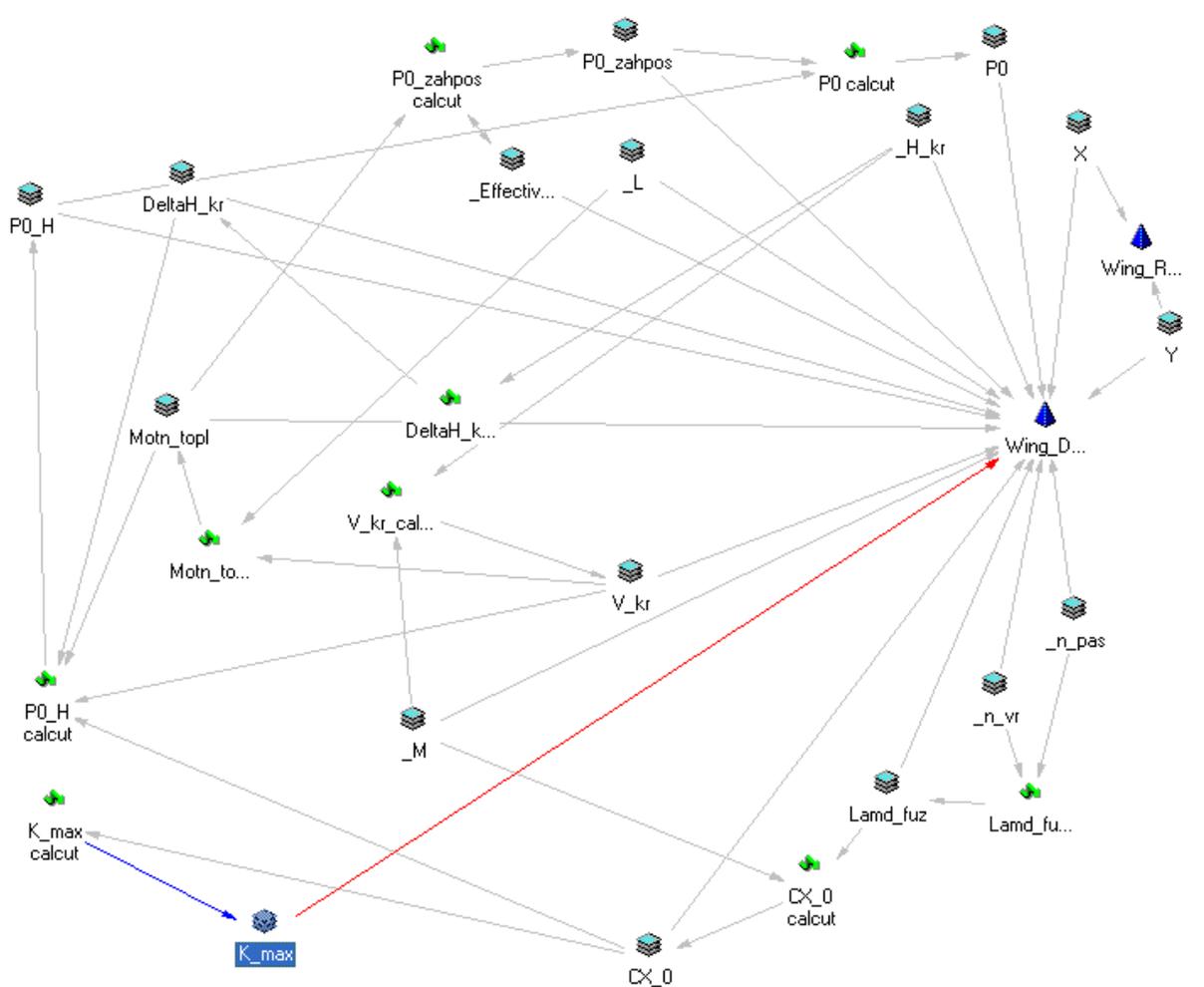


Рисунок 2.12 – Представление онтологии предметной области “Выбор удельной нагрузки на крыло” в виде семантической сети

## 2.1.5. Проектирование онтологии мира заказов и ресурсов

### 2.1.5.1 Создание онтологии мира заказов и ресурсов

Для создания онтологии мира заказов и ресурсов (онтологии виртуального мира) необходимо выделить библиотеку онтологий и в контекстном меню выбрать пункт *New item -> Virtual world ontology*. При

этом в правой части экрана появится диалог создания онтологии мира заказов и ресурсов, предоставляющий возможность выбора тех концептов «объект», для которых требуется создание агентов заказа или ресурса. Если предполагается, что какой-либо объект должен иметь одновременно и агента заказа, и агента ресурса, следует выставить флажок слева от названия этого объекта. Агенты будут созданы автоматически. Если объекту в виртуальном мире должен соответствовать агент заказа, либо агент ресурса, флажок выставлять не следует. Агенты будут созданы позже в индивидуальном порядке. Так, в данном примере объект «проектируемый самолет» выступает в роли заказа и должен иметь в виртуальном мире только агента заказа. В свою очередь, объект-ресурс выступает в роли ресурса и должен иметь в виртуальном мире только агента ресурса. Нажатие на кнопку <OK> подтверждает необходимость создания онтологии виртуального мира (рис. 2.13).

При нажатии <OK> пиктограмма онтологии виртуального мира появляется в архитектуре онтологий в левой части экрана. При нажатии <+> раскрывается дерево концептов онтологии виртуального мира, которое содержит концепты агентов заказов и ресурсов, а также отношений между агентами (рис. 2.13).

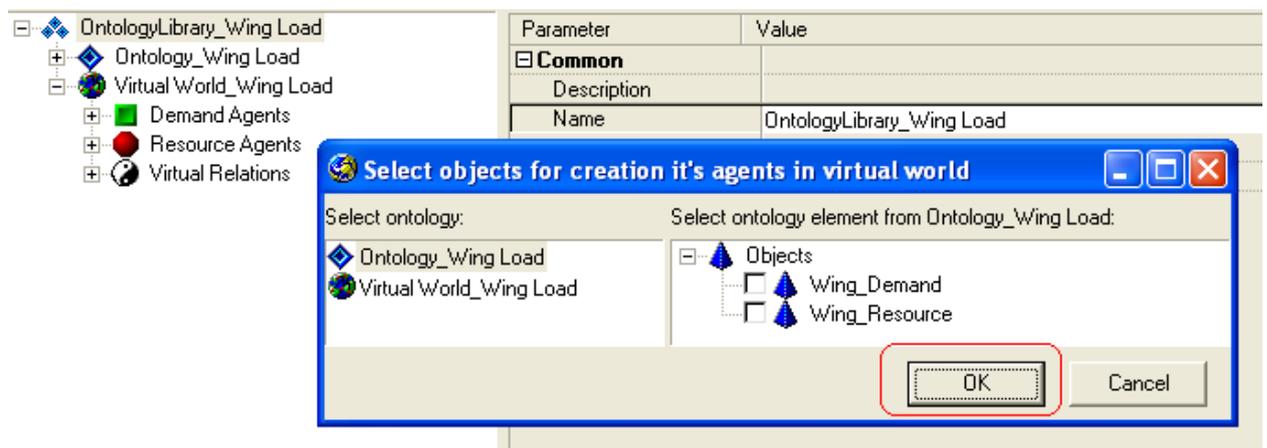


Рисунок 2.13 – Выбор объектов, для которых должны создаваться агенты

Создайте онтологию мира заказов и ресурсов для предметной области «Выбор удельной нагрузки на крыло» (*New Item -> Virtual World Ontology*). Введите в поле *Name* название онтологии - *Virtual World\_Wing Load*. Раскройте дерево концептов онтологии виртуального мира (рис. 2.14).

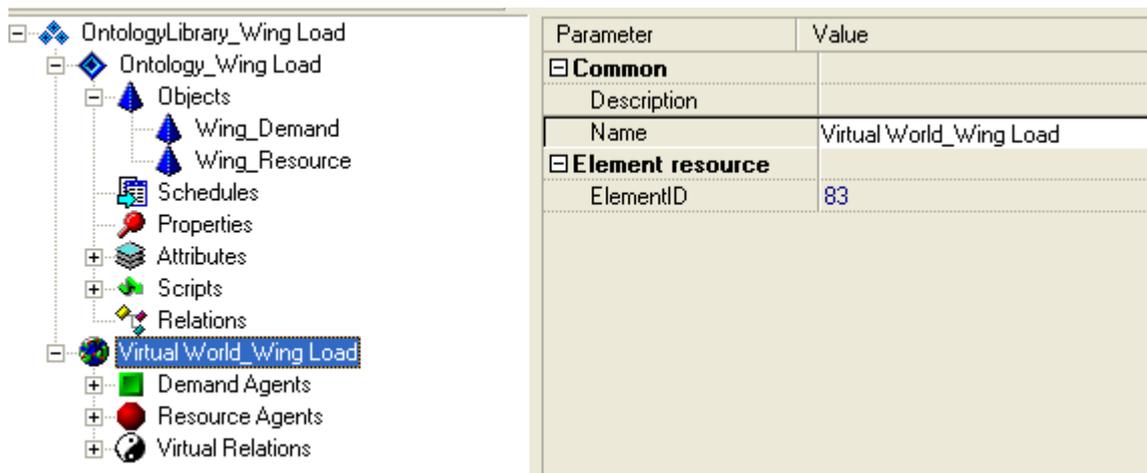


Рисунок 2.14 – Категории концептов онтологии мира заказов и ресурсов

### 2.1.5.2 Создание концепта «агент заказа»

Создайте концепт «агент заказа» для концепта «проектируемый самолет» (т. к. именно проект является активной сущностью): выделите категорию *Demand Agents*, в контекстном меню выберите пункт *New Item -> Demand agent* и в появившемся диалоге выберите концепт *Wing\_Demand*. Затем нажмите *<OK>* и переименуйте созданный концепт в *Wing\_Demand* (рис. 2.15). Назначьте концепту «агент заказа» три вида пиктограмм, с помощью которых данный концепт будет отображаться при работе со сценой в процессе моделирования. Установите у концепта флаг *vaoAutoCreate* (должен быть установлен по умолчанию).

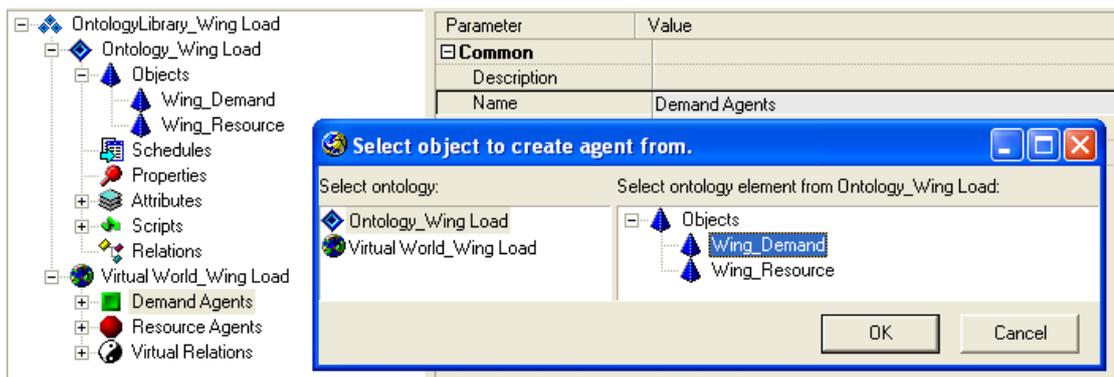


Рисунок 2.15 – Создание агента заказа для концепта *Wing\_Demand*

### 2.1.5.3 Создание концепта «агент ресурса»

Создайте концепт «агент ресурса» для концепта «объект-ресурс»: выделите категорию *Resource Agents*, в контекстном меню выберите пункт *New Item -> Resource agent* и в появившемся диалоге выберите концепт *BD\_Plane*. Затем нажмите *<OK>* и переименуйте созданный концепт в *BD\_Plane Resource* (рис. 2.16). Назначьте концепту «агент ресурса» три типа пиктограмм, с помощью которых данный концепт будет отображаться при

работе со сценой в процессе моделирования. Установите у концепта флаг *vaoAutoCreate*. Не устанавливайте у концепта флаг *raoActive*.

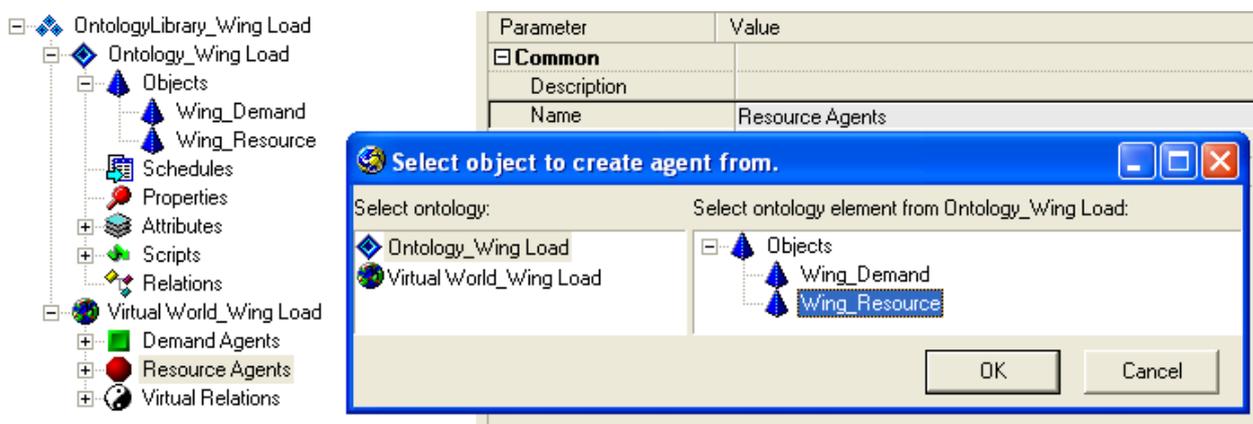


Рисунок 2.16 – Создание агента ресурса для концепта *Wing\_Resource*

#### 2.1.5.4 Виртуальные отношения: отношения матчинга

Отношение матчинга является служебным классом отношений в виртуальном мире и связывает между собой концепты заказов/ресурсов. Отношение матчинга показывает *возможность* матчинга между агентами, концепты которых в онтологии связаны данным отношением. Иными словами, матчинг возможен, но он не обязательно состоится: агенты могут не договориться по разным причинам (есть более выгодное предложение, данное предложение не устраивает партнера/агента и т.д.).

Отношение матчинга возможно только между агентами заказа и ресурса. Например, матчинг агента заказа с агентом другого заказа невозможен. Отношение матчинга является отношением вида «субъект-объект». Субъект выступает инициатором матчинга. Агент заказа и агент ресурса могут устанавливать отношение матчинга в сцене, причем инициатором (субъектом) матчинга может выступать как агент заказа, так и агент ресурса (если для него установлен флаг активности *raoActive*).

Установите отношение одностороннего матчинга между концептами агента заказа *Wing\_Demand* и агента ресурса *Wing\_Resource*. Для этого в категории *Virtual Relations* выделите концепт *Matching relation* и в контекстном меню выберите *Establish relation*. В правом окне раскройте дерево агентов виртуального мира, а в нем - категории *DemandAgents* и *ResourceAgents*. В качестве *Matching subject* выберите *Wing\_Demand*, а в качестве *Matching object* – *Wing\_Resource* (рис. 2.17). Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Wing\_Demand.Wing\_Resource* (рис. 2.17).

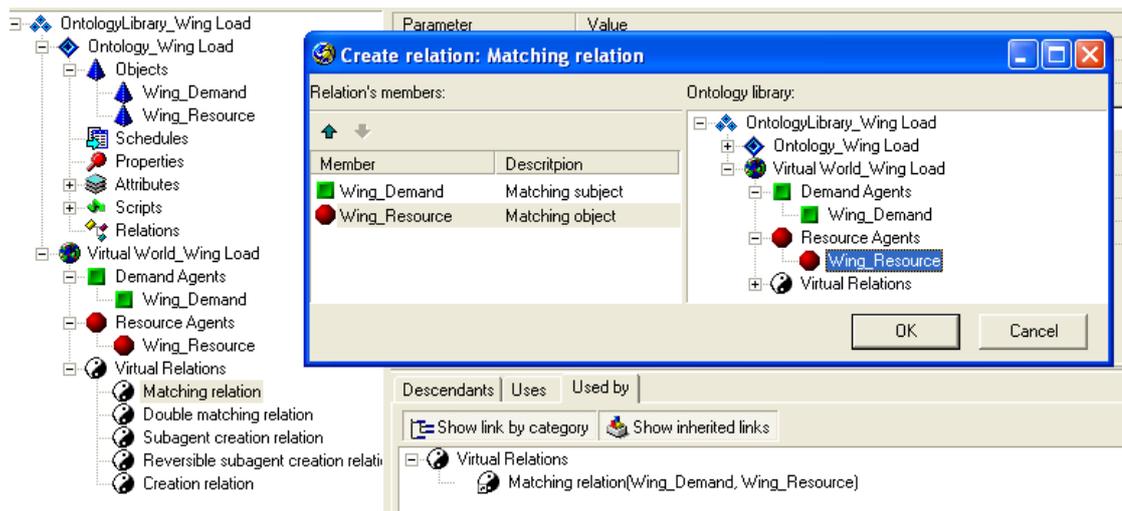


Рисунок 2.17 – Связывание отношением матчинга концептов *Wing\_Demand* и *Wing\_Resource*

### 2.1.5.5 Условия матчинга

Перейдите в закладку *Used by* (*Virtual relations* -> *Matching relation* -> *Used by*) и выделите *Matching relation (Wing\_Demand, Wing\_Resource)*, далее в контекстном меню выберите *Edit virtual relation properties* (рис. 2.18). При этом откроется окно редактирования условий матчинга (рис. 2.19).

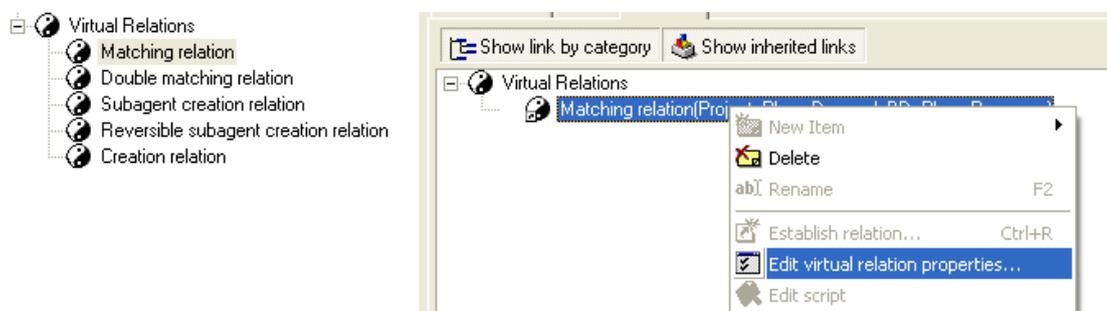


Рисунок 2.18 – Редактирование свойств отношения матчинга

В окне редактирования условий матчинга имеются следующие закладки:

- *Matching conditions* – создание и редактирование условий матчинга. Знаковое и скриптовое условия описаны выше. *Name* – тип условия (записывается автоматически), *Checking agent* – агент проверяющий условие матчинга, т.е. агент-субъект (записывается автоматически).
- *Decision Making Machine conditions* – создание и редактирование критериев, на основании которых в процессе матчинга принимается решение о резервировании агентом заказа агента ресурса.
- *Tasks* – формирование заданий на расчет дополнительных атрибутов, необходимых матчеру (в данной работе не используется).

- *Events* – обработчик событий, которые используются, когда необходимо изменить значение какого-либо атрибута агента в зависимости от значения атрибутов в матче (в данной работе не используется).



Рисунок 2.19 – Окно редактирования условий матчинга

Так как в данной задаче матчинг необходим только для того, чтобы запустить скрипты, выполняющие расчеты, никаких ограничений не существует и никаких особых условий матчинга создавать не следует.

#### 2.1.5.6 Условия принятия (*Decision Making Machine conditions*)

Условия принятия решения предназначены для работы машины принятия решений и позволяют агенту выбрать одно из множества возможных предложений (матчингов) от других партнёров. Условие задаётся в закладке *Decision Making Machine conditions* окна редактирования *Edit Matching Conditions* при помощи кнопки . Для условия принятия решения необходимо определить атрибут условия, направление оптимизации (максимум/минимум) и весовой коэффициент, определяющий «значимость» данного условия.

- **Создание условия принятия решения – выбор минимального из значений удельной нагрузки на крыло при заходе на посадку и удельной нагрузки на крыло при заданной крейсерской скорости на расчетной высоте полета**

За расчетное значение удельной нагрузки на крыло согласно формуле (2.11) следует принять минимальное из двух значений: удельной нагрузки на крыло при заходе на посадку (формула (2.1)) и удельной нагрузки на крыло при заданной крейсерской скорости на расчетной высоте полета (формула (2.6)).

Создайте условие 1 принятия решений для матчинга *Wing\_Demand -> Wing\_Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия (рис. 2.20):

- Attribute = ‘Wing\_Demand.P0’;
- Order = ‘Min’;
- Weight = ‘100’.

Активизируйте условие принятия решения, выставив флажок *Active*.

– **Создание условия принятия решения - максимизация аэродинамического качества**

Создайте условие 2 принятия решений для матчинга *Wing\_Demand* -> *Wing\_Resource*. Укажите следующие параметры условия (рис. 2.20):

- Attribute = ‘Wing\_Demand.K\_max’;
- Order = ‘Max’;
- Weight = ‘100’.

Активизируйте условие принятия решения.

Условия принятия решения в матчинге для выполнения расчетов приведены на рис. 2.20.

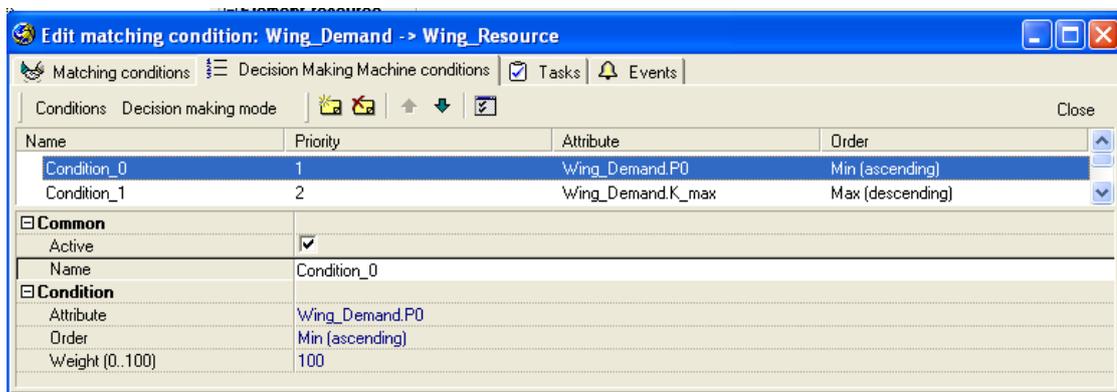


Рисунок 2.20 – Условия принятия решения в матчинге *Wing\_Demand* – *Wing\_Resource*

### 2.1.5.7 Параметры представления экземпляра объекта в сцене

Поведение экземпляра концепта «объект» (агента) в сцене определяет Группа свойств концепта «объект» *Interface behaviour* (рис. 2.21).

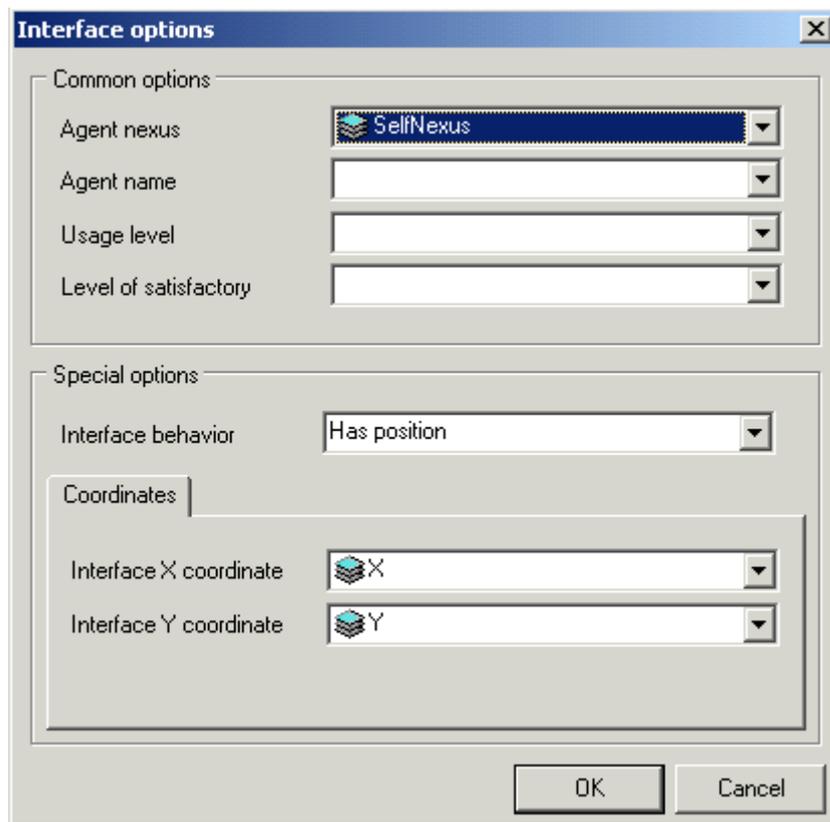


Рисунок 2.21 – Редактор свойств *Interface behaviour*

Свяжите атрибуты *X*, *Y* с позицией агента объекта *Wing\_Demand* в сцене. Для этого в редакторе свойств объекта *Wing\_Demand* вызовите диалог редактирования свойства *Interface behaviour*, в списке атрибутов этого свойства выберите параметр *Has position*. Затем в качестве *Interface X coordinate* и *Interface Y coordinate* выберите атрибуты *X* и *Y* соответственно. Нажмите <OK>. Свяжите атрибуты *X*, *Y* с позицией агента объекта *Wing\_Resource* в сцене. Для этого в редакторе свойств объекта *Wing\_Resource* вызовите диалог редактирования свойства *Interface behaviour*, в списке атрибутов этого свойства выберите параметр *Has position*. Затем в качестве *Interface X coordinate* и *Interface Y coordinate* выберите атрибуты *X* и *Y* соответственно. Нажмите <OK>.

#### **2.1.5.8 Сохранение онтологий предметной области “Выбор удельной нагрузки на крыло”**

Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор удельной нагрузки на крыло” приведено на рис. 2.22.

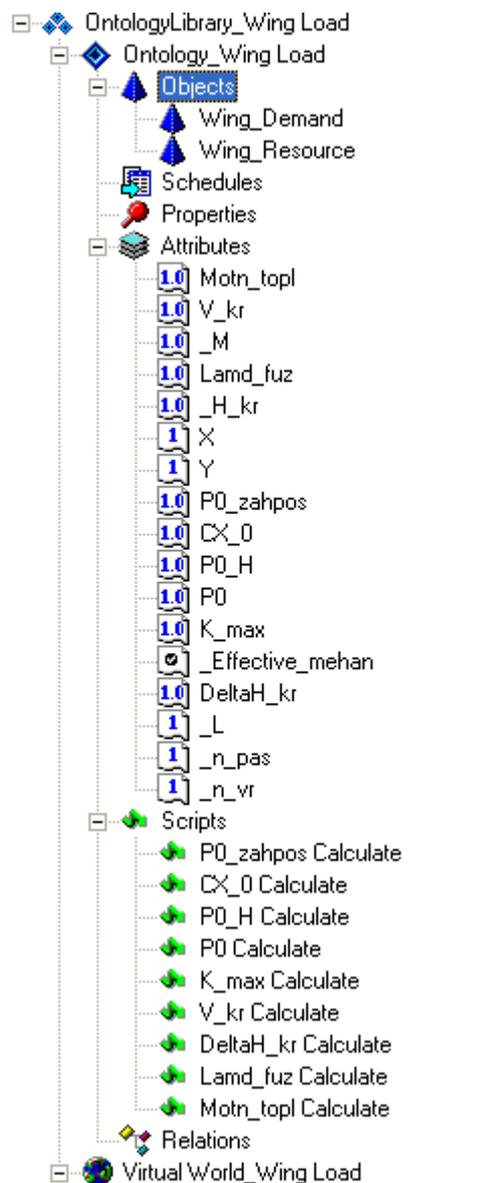


Рисунок 2.22 – Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор удельной нагрузки на крыло”

Сохраните созданные онтологии (дескриптивную онтологию и онтологию мира заказов/ресурсов) при помощи кнопки  под именем *Wing Load Ontology*. Расширение *.ocl* будет добавлено автоматически. По умолчанию, файл онтологии будет размещен в разделе *Ontology Samples*. Завершите работу с конструктором онтологий (*File -> Close*).

### 2.1.6. Создание онтологической сцены

Выполните на Вашем компьютере следующие настройки: *Пуск -> Панель управления -> Язык и региональные стандарты -> Настройка -> в поле “Разделитель целой и дробной части” установите точку (рис. 23).*

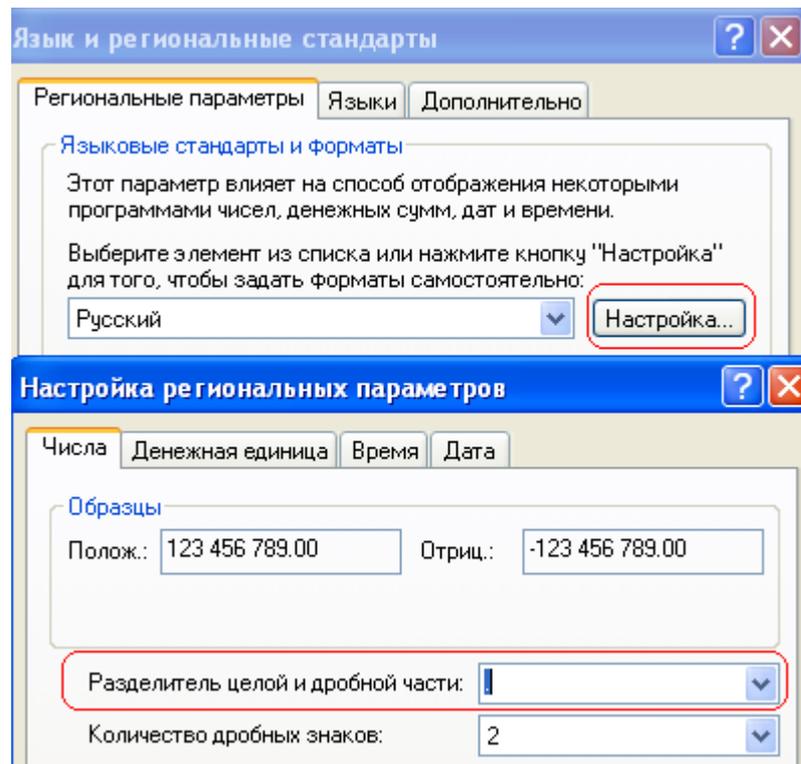


Рисунок 2.23 – Настройки компьютера

Вызовите исполняющую систему, запуская программу  `Unifntf.exe`, находящуюся в папке `OntConsUnifntf`. Создайте новую онтологическую сцену (`File` → `New scene` → `Load ontology`, выберите онтологию `Wing Load ontology.ocl`). В окне физического мира создайте одного агента проектируемого самолета `Wing Demand_1` и одного агента объекта-ресурса `Wing Resource_1` для выполнения расчетов (рис.2.24). С помощью инспектора агентов установите следующие значения атрибутов для агентов, например, для самолета ТУ-154.

<i>Атрибуты</i>	<i>Ту-154</i>	<i>Ил-86</i>	<i>Як-40</i>
<b><code>_Effective_mehan</code></b>	+	+	+
<b><code>_H_kr</code></b>	10100	10000	8100
<b><code>_M</code></b>	0.88	0.88	0.47
<b><code>_L</code></b>	5200	4350	1350
<b><code>_n_pas</code></b>	164	350	32
<b><code>_n_vr(3..10)</code></b>	6	8	3

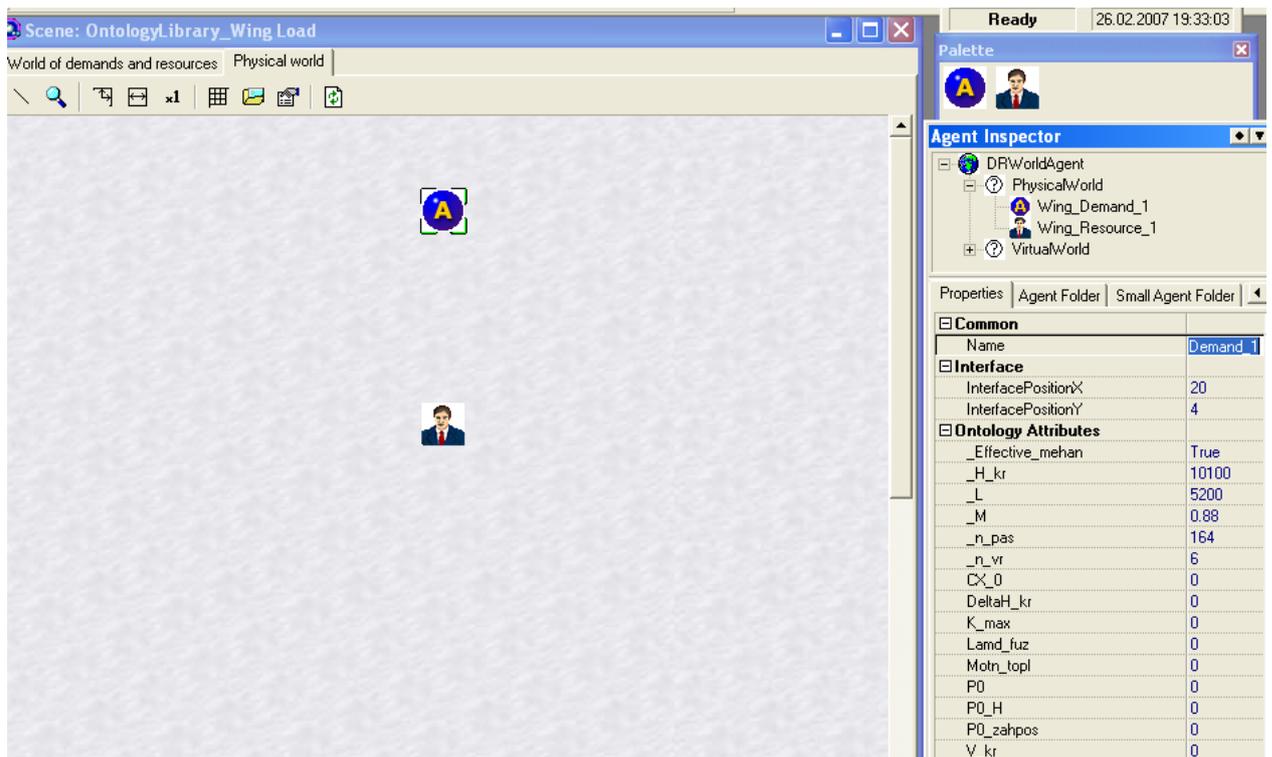


Рисунок 2.24 – Создание онтологической сцены предметной области «Выбор удельной нагрузки на крыло»

## 2.1.7. Моделирование сцены виртуального мира

### 2.1.7.1. Запуск сцены на моделирование для самолета Ту-154

Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*). При помощи кнопки  запустите сцену на моделирование (т.е., запустите выполнение процесса матчинга). Наблюдайте процесс матчинга между агентами проектируемого самолета и агентами самолетов-прототипов в базе данных.

В процессе матчинга активный агент проектируемого самолета запускает скрипты, необходимые для вычисления значений атрибутов. Далее агент проекта строит таблицу принятия решений, в которой указан единственный зарезервированный агент объекта-ресурса. По двойному клику левой кнопкой мыши на пиктограмме агента *Wing Demand\_1* в окне физического или виртуального мира (либо при выполнении в контекстном меню агента команд *Actions -> Show agent structure*) можно открыть таблицу, в которой представлена структура агента (в закладке *Matchers*). В таблице указывается список ресурсов (партнеров), т.е. агентов, которые зарезервированы агентом-проектом (рис. 25). При выделении партнера в правом окне отображаются атрибуты агента-проекта (простые (*Simple*), значения которых задавались непосредственно в сцене, и вычисляемые (*Scripted*), значения которых рассчитывались в процессе матчинга с помощью скриптов).

Partner	Status	Name	Value	State	Type
Wing_Resource_1	Accept	P0	583.295401338241	Assigned	Scripted
		K_max	16.8443049504479	Assigned	Scripted
		P0_zahpos	583.295401338241	Assigned	Scripted
		_Effective_mehan	True	Assigned	Scripted
		Motn_topl	0.420738113266803	Assigned	Scripted
		V_kr	263.204999703854	Assigned	Scripted
		_H_kr	10100	Assigned	Simple
		_M	0.88	Assigned	Simple
		_L	5200	Assigned	Simple
		P0_H	948.917869693771	Assigned	Scripted
		CX_D	0.0215435219609484	Assigned	Scripted
		Lamd_fuz	8.24833110814419	Assigned	Scripted
		_n_vr	6	Assigned	Simple
		_n_pas	164	Assigned	Simple
		DeltaH_kr	0.3354	Assigned	Scripted

Рисунок 2.25 – Структура агента проекта *Wing Demand\_1* самолета ТУ-154

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 26), в которой перечислены основные параметры, вычисляемые с помощью скриптов).

Agent	P0	K_max
Wing_Resource_1	583.295401338241	16.8443049504479

Рисунок 2.26 – Таблица принятия решений агента *Wing Demand\_1* самолета ТУ-154

Агент-ресурс в данной задаче не имеет атрибутов, поэтому таблица, содержащая его структуру, пуста.

Окончательные результаты матчинга можно видеть на рисунке 27. В результате матчинга была выполнена следующая операция резервирования: – *Wing Demand\_1* – *Wing Resource\_1*.

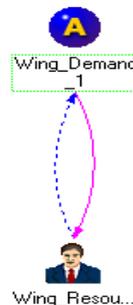


Рисунок 2.27 – Результаты матчинга

### 2.1.7.2. Сохранение сцены виртуального мира

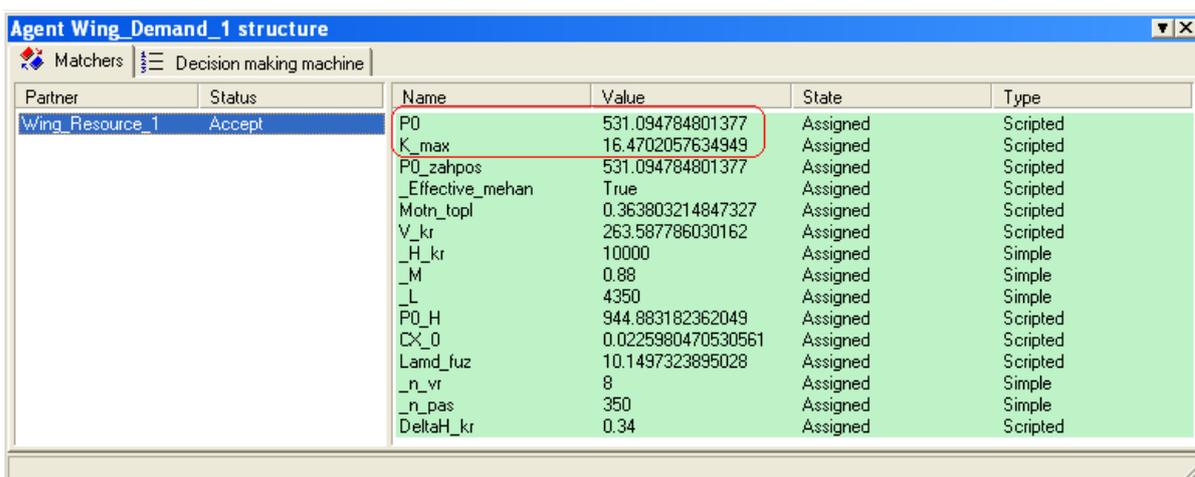
Для сохранения сцены используются средства онтологического сохранения: кнопка  или последовательность команд *Tools - > Save ontology scene...* Сохраните сцену под именем *Scene\_Wing Load TY-154*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*. Завершите работу с исполняющей системой (*File -> Close*).

### 2.1.7.3. Загрузка ранее созданной сцены виртуального мира

Для загрузки ранее созданной сцены используются средства онтологической загрузки: кнопка  или последовательность команд *Tools - > Load ontology scene...* Загрузите сцену *Scene\_Wing Load TY-154.osf*.

### 2.1.7.4. Моделирование сцены для самолета Ил-86

Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*). У агента проекта *Wing Demand\_1* измените значения атрибутов. Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*). При помощи кнопки  запустите сцену на моделирование (т.е., запустите процесс матчинга). Наблюдайте процесс матчинга между агентами проекта и самолетов-прототипов. Результаты матчинга показаны на рис. 2.28, а таблица принятия решения агента проекта – на рис. 2.29.



Partner	Status	Name	Value	State	Type
Wing_Resource_1	Accept	P0	531.094784801377	Assigned	Scripted
		K_max	16.4702057634949	Assigned	Scripted
		P0_zahpos	531.094784801377	Assigned	Scripted
		_Effective_mehan	True	Assigned	Scripted
		Motr_topl	0.363803214847327	Assigned	Scripted
		V_kr	263.587786030162	Assigned	Scripted
		_H_kr	10000	Assigned	Simple
		_M	0.88	Assigned	Simple
		_L	4350	Assigned	Simple
		P0_H	944.883182362049	Assigned	Scripted
		CX_0	0.0225980470530561	Assigned	Scripted
		Lamd_fuz	10.1497323895028	Assigned	Scripted
		_n_vr	8	Assigned	Simple
		_n_pas	350	Assigned	Simple
		DeltaH_kr	0.34	Assigned	Scripted

Рисунок 2.28 – Структура агента проекта *Wing Demand\_1* самолета ИЛ-86

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 29), в которой перечислены основные параметры, вычисляемые с помощью скриптов).

Agent	P0	K_max
Wing_Resource_1	531.094784801377	16.4702057634949

Рисунок 2.29 – Таблица принятия решений агента *Wing Demand\_1* ИЛ-86

Агент-ресурс в данной задаче не имеет атрибутов, поэтому таблица, содержащая его структуру, пуста. При помощи средств онтологического сохранения сохраните сцену под именем *Scene\_Wing Load ИЛ-86*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*. Завершите работу с исполняющей системой (*File -> Close*).

### 2.1.7.5. Моделирование сцены для самолета Як-40

Аналогичным образом запустите процесс матчинга для самолета Як-40. Результаты матчинга показаны на рис. 2.30, а таблица принятия решения агента проекта – на рис. 2.31.

Partner	Status	Name	Value	State	Type
Wing_Resource_1	Accept	P0	301.692101612969	Assigned	Scripted
		K_max	17.7222838213424	Assigned	Scripted
		P0_zahpos	443.6284347616	Assigned	Scripted
		_Effective_mehan	True	Assigned	Scripted
		Motn_topl	0.23836984235792	Assigned	Scripted
		V_kr	144.610533125453	Assigned	Scripted
		_H_kr	8100	Assigned	Simple
		_M	0.47	Assigned	Simple
		_L	1350	Assigned	Simple
		P0_H	301.692101612969	Assigned	Scripted
		CX_0	0.0193406875205497	Assigned	Scripted
		Lamd_fuz	5.89634864546525	Assigned	Scripted
		_n_vr	3	Assigned	Simple
		_n_pas	32	Assigned	Simple
		DeltaH_kr	0.4274	Assigned	Scripted

Рисунок 2.30 – Структура агента проекта *Wing Demand\_1* самолета Як-40

Таблица принятия решений агента проекта представлена на рис. 2.31.

Agent	P0	K_max
Wing_Resource_1	301.692101612969	17.7222838213424

Рисунок 2.31 – Таблица принятия решений агента *Wing Demand\_1* Як-40

Сохраните сцену под именем *Scene\_Wing Load Як-40* и завершите работу с исполняющей системой.

## 1.2. Часть 2. Выбор тяговооруженности проектируемого самолета

### 2.2.1. Выбор значений потребной тяговооруженности самолета

#### 2.2.1.1. Расчет тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости полета на расчетной высоте

Значение тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте полета определяется по формуле (2.12)

$$\overline{P}_0^I = \frac{1}{\xi \cdot \Delta H_{кр}^{0,85} \cdot 0,85 \cdot K_{крейс}}, \quad (2.12)$$

где:

- $\overline{P}_0^I$  - тяговооруженность из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте полета;
- $K_{крейс}$  – аэродинамическое качество на крейсерском режиме;
- $\Delta H_{кр}$  - относительная плотность воздуха на крейсерской высоте полета;
- $\xi$  - коэффициент, учитывающий изменение тяги двигателя по скорости полета.

Аэродинамическое качество на крейсерском режиме рассчитывается по формуле (2.13)

$$K_{крейс} = 0,8 * K_{max}, \quad (2.13)$$

где:

- $K_{max}$  – максимальное аэродинамическое качество (рассчитывается в первой части лабораторной работы №2).

Относительная плотность воздуха на крейсерской высоте полета рассчитывается по формуле (3)

$$\Delta H_{кр} = 0,8 - 0,000046 H_{кр}, \quad (2.13)$$

где  $H_{кр}$  – крейсерская высота полета, м (исходное данное).

Коэффициент, учитывающий изменение тяги двигателя по скорости полета, рассчитывается по формуле (4)

$$\xi_1 = 1 - 0,32M + 0,4M^2 - 0,01M^3, \quad (2.14)$$

где  $M$  - число Маха (исходное данное).

### 2.2.1.2. Расчет тяговооруженности из условия заданной длины разбега

Тяговооруженность из условия обеспечения заданной длины пробега определяется по формуле (2.15)

$$\overline{P}_0^{II} = \frac{1,26 p_0}{C_{y \max \text{ взл}} \cdot L_{\text{разб}}} + 0,09975 \quad , \quad (2.15)$$

где:

- $p_0$  - удельная нагрузка на крыло (рассчитывается в лабораторной работе №2),  $\text{даН/м}^2$  ;
- $L_{\text{разб}}$  - длина разбега по ВПП (исходное данное), м;
- $C_{y \max \text{ взл}}$  - значение максимального коэффициента подъемной силы самолета во взлетной конфигурации; значение  $C_{y \max \text{ взл}}$  выбирается в зависимости от системы механизации крыла:
  - для эффективной механизации  $C_{y \max \text{ взл}} = 2.5$ ;
  - для слабой (неэффективной) механизации  $C_{y \max \text{ взл}} = 2.0$ .

### 2.2.1.3. Расчет тяговооруженности из условия взлета при отказе одного двигателя

Тяговооруженность из условия взлета при отказе одного двигателя определяется по формуле (6)

$$\overline{P}_0^{III} = \frac{1,5 \cdot n_{\text{дв}}}{n_{\text{дв}} - 1} (0,083 + \text{tg} \theta_{\min}) \quad , \quad (2.16)$$

где:

- $n_{\text{двиг}}$  – количество двигателей на самолете, шт (исходное данное);
- $\text{tg} \theta_{\min}$  - градиент набора высоты на взлетном режиме, радиан.

Градиент набора высоты на взлетном режиме определяется в зависимости от количества двигателей следующим образом:

Количество двигателей	Градиент набора высоты
2	0.024
3	0.027
>3	0.030

### 2.2.1.4. Расчет потребной тяговооруженности самолета

Потребная тяговооруженность самолета, приведенная к взлетному режиму, определяется по формуле (7)

$$P = \max (\overline{P}_0^I, \overline{P}_0^{II}, \overline{P}_0^{III}). \quad (2.17)$$

где:

- $\overline{P}$  – расчетное значение потребной тяговооруженности ;
- $\overline{P}_0^I$  - тяговооружённость из условия обеспечения горизонтального полета на крейсерской скорости полета  $V_{\text{крейс}}$  на расчетной высоте  $H_{\text{крейс}}$  определяется по формуле (2.12);
- $\overline{P}_0^{II}$  - тяговооружённость из условия заданной длины разбега, рассчитывается по формуле (2.15);
- $\overline{P}_0^{III}$  - тяговооруженность из условия взлета при отказе одного двигателя определяется по формуле (2.16).

## 2.2.2. Проектирование дескриптивной онтологии

### 2.2.2.1. Создание онтологии

Загрузите конструктор онтологий (файл *OntCons.exe*). Создайте новую библиотеку онтологий (*File* → *New*). Переименуйте ее, вводя в поле *Name* имя *OntologyLibrary\_Draught*. Создайте дескриптивную онтологию предметной области «Выбор потребной тяговооруженности самолета» (*New Item* → *Descriptive ontology*). Переименуйте ее, вводя в поле *Name* имя *Ontology\_Draught*.

### 2.2.2.2. Создание онтологии

#### – Концепт «объект»

После создания дескриптивной онтологии ПО «Выбор потребной тяговооруженности самолета» необходимо создать два концепта «объект». Имена атрибутов, которые являются исходными данными для расчета, начинаются с нижнего подчеркивания.

Создайте объект *Demand\_Draught* и объект *Resource\_Draught* (*Objects* → *New Item* → *Object*), назначьте этим объектам пиктограммы, соответствующие трем видам отображения. Каждый концепт «объект» может иметь определенный список атрибутов.

## – Концепт «атрибут»

Концепт «атрибут» – это величина, характеризующая объект (количественное выражение признака).

- Создайте **булевский атрибут** *\_Effective\_mehan* (*Attributes -> New Item -> Boolean Attribute*), переименуйте созданный атрибут в *\_Effective\_mehan*.
- Создайте **вещественные атрибуты** (*Attributes -> New Item -> Float Attribute*), переименуйте созданные атрибуты в:
  - *\_H\_kr*,
  - *\_M*,
  - *\_P*,
  - *\_Dist\_razb*,
  - *\_K\_max*,
  - *DeltaH\_kr*,
  - *Ksi*,
  - *K\_kreis*,
  - *TG*,
  - *P0\_1*,
  - *P0\_2*,
  - *P0\_3*,
  - *P0\_max*,
  - *Temp*.
- Создайте **целые атрибуты** (*New Item -> Integer Attribute*), переименуйте созданные атрибуты в:
  - *\_N\_dvig*,
  - *X*,
  - *Y*.

### 2.2.2.3. Создание связей между концептами

Установите связи между созданными концептами дескриптивной онтологии ПО «Выбор потребной тяговооруженности самолета», т.е. укажите, что объект *Demand\_Draught* имеет атрибуты *\_Effective\_mehan*, *\_H\_kr*, *\_M*, *\_P*, *\_Dist\_razb*, *\_N\_dvig*, *\_K\_max*, *DeltaH\_kr*, *Ksi*, *K\_kreis*, *TG*, *P0\_1*, *P0\_2*, *P0\_3*, *P0\_max*, *Temp*, *X*, *Y*, а объект *Resource\_Draught* – атрибуты *X*, *Y*.

Просмотреть список атрибутов данного объекта можно в редакторе свойств концепта «объект».

В результате, у объекта в закладке *Uses* появится список имен атрибутов, а у атрибута в закладке *Used by* появится имя объекта (объектов), использующего этот атрибут. На рис. 2.32 показаны свойства объекта *Demand\_Draught* в закладке *Uses*, причем данный объект имеет атрибуты

*\_Effective\_mehan, \_H\_kr, \_M, \_P, \_Dist\_razb, \_N\_dvig, \_K\_max, DeltaH\_kr, Ksi, K\_kreis, TG, P0\_1, P0\_2, P0\_3, P0\_max, Temp, X, Y* (соответствующие связи отображены в закладке *Uses*). В закладке *Used by* атрибутов отображена их связь с соответствующим объектом.

Parameter	Value
<b>Common</b>	
Description	
ElementType	Object
Name	Demand_Draught
<b>Element resource</b>	
ElementID	70
Extra large icon (50x50)	(TBitmap)
Large icon (32x32)	(TBitmap)
Small icon (16x16)	(TBitmap)
<b>Interface options</b>	
Add to palette	<input checked="" type="checkbox"/>
Interface behavior	<interface behavior properties>

Descendants Uses **Used by**

Show link by category Show inherited links

- Attributes
  - \_M
  - Ksi
  - \_K\_max
  - P0\_1
  - \_P
  - \_Dist\_Razb
  - P0\_2
  - \_N\_dvig
  - TG
  - P0\_3
  - P0\_max
  - Temp
  - X
  - Y
  - \_Effective\_mehan
  - \_H\_kr
  - DeltaH\_kr
  - K\_kreis

Рисунок 2.32 – Атрибуты (свойства) объекта *Demand\_Draught* в закладке *Uses*

## – Концепт «скрипт»

Вызовите редактор скриптов и напишите, или отредактируйте тело скрипта (вызвать *Other -> Script*  в редакторе свойств скрипта, либо нажать кнопку  Edit script в закладке *Script body*).

– **Определение скрипта для вычисления аэродинамического качества на крейсерском режиме**

Чтобы вычислить с помощью скрипта аэродинамическое качество на крейсерском режиме по формуле (2.12), необходимо создать концепт «скрипт». Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*. Созданный скрипт переименуйте в *K\_kreis Calculate* и свяжите его с атрибутом *K\_kreis*, т.е. перетащите скрипт на атрибут *K\_kreis*. Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметром скрипта *K\_kreis Calculate*, вычисляющего аэродинамическое качество на крейсерском режиме, является атрибут *\_K\_max* (рис. 2.33). Напишите тело скрипта. Для этого выделите скрипт *K\_kreis Calculate* в дереве концептов дескриптивной онтологии, затем перейдите в закладку *Script body*, нажмите на кнопку . При этом откроется окно редактора скриптов, в котором необходимо набрать следующий текст (имена концептов заключаются в кавычки, незначащие пробелы в начале идентификаторов не допускаются). Имена концептов необходимо выбирать в списке концептов, которые являются параметрами скрипта. Скрипт заканчивается точкой с запятой.

**begin**

```
result:= 0.8 * "_K_max";
```

**end;**

Сохраните скрипт нажатием на кнопку . Закройте окно редактора скриптов. Проверьте правильность синтаксиса скрипта нажатием на кнопку *Check syntax*. Если будет зафиксирована ошибка, необходимо вызвать редактор скриптов и внести необходимые исправления.

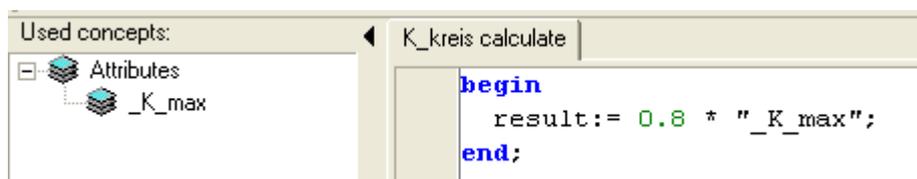


Рисунок 2.33 – Атрибуты и тело скрипта *K\_kreis Calculate*

Аналогичным способом создайте и другие скрипты.

– **Определение скрипта для вычисления относительной плотности воздуха на расчетной высоте** (по формуле 2.13).

Созданный скрипт переименуйте в *DeltaH\_kr Calculate* и свяжите его с атрибутом *DeltaH\_kr*. Параметром скрипта *DeltaH\_kr Calculate*,

вычисляющего относительную плотность воздуха на расчетной высоте, является атрибут *\_H\_kr*. Напишите тело скрипта (рис. 2.34).

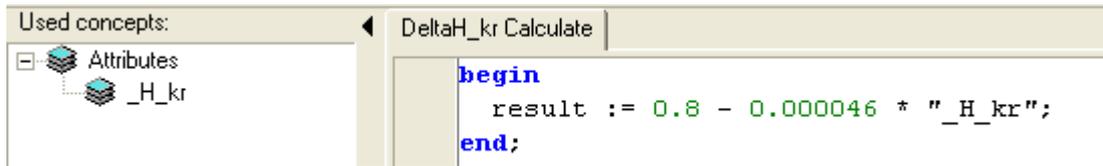


Рисунок 2.34 – Атрибуты и тело скрипта *DeltaH\_kr Calculate*

- **Определение скрипта для вычисления коэффициента, учитывающего изменение тяги двигателя по скорости полета** (по формуле 2.14).

Созданный скрипт переименуйте в *Ksi Calculate* и свяжите его с атрибутом *Ksi*. Параметром скрипта *Ksi Calculate*, вычисляющего коэффициент, учитывающий изменение тяги двигателя по скорости полета, является атрибут *\_M*. Напишите тело скрипта (рис. 2.35).

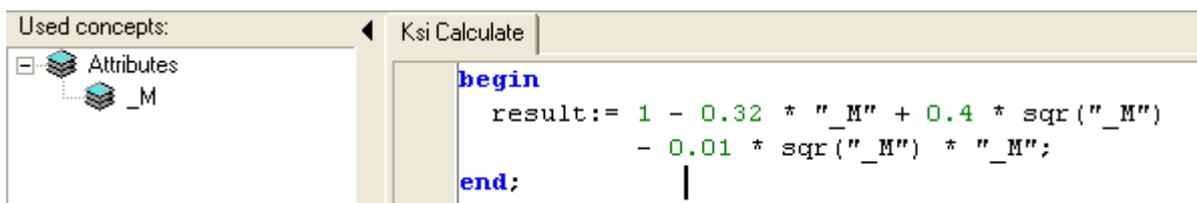


Рисунок 2.35 – Атрибуты и тело скрипта *Ksi Calculate*

- **Определение скрипта для вычисления тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте** (по формуле 2.12).

Созданный скрипт переименуйте в *P0\_1 Calculate* и свяжите его с атрибутом *P0\_1*. Параметрами скрипта *P0\_1 Calculate*, вычисляющего тяговооруженность из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте, являются атрибуты *Ksi*, *K\_kreis* и *DeltaH\_kr*. Напишите тело скрипта (рис. 2.36).

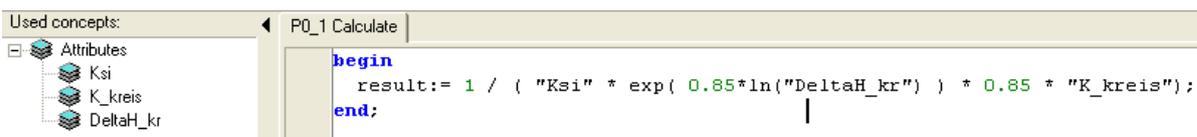


Рисунок 2.36 – Атрибуты и тело скрипта *P0\_1 Calculate*

- **Определение скрипта для вычисления тяговооруженности из условия обеспечения заданной длины разбега** (по формуле 2.15).

Созданный скрипт переименуйте в *PO\_2 Calculate* и свяжите его с атрибутом *PO\_2*. Параметрами скрипта *PO\_2 Calculate*, вычисляющего тяговооруженность из условия заданной длины разбега, являются атрибуты *\_P*, *Dist\_razb*, *\_Effective\_mehan*. Напишите тело скрипта (рис. 2.37).

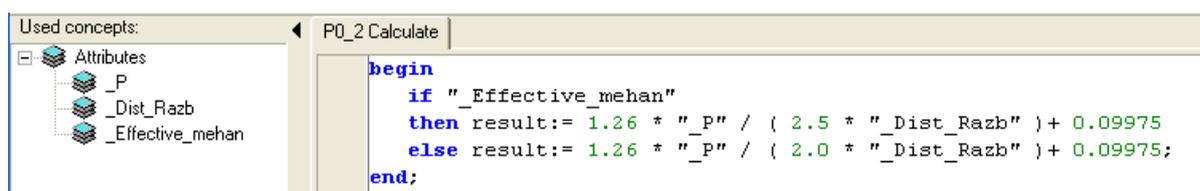


Рисунок 2.37 – Атрибуты и тело скрипта *PO\_2 Calculate*

- **Определение скрипта для вычисления градиента набора высоты при взлетном режиме**

Созданный скрипт переименуйте в *TG Calculate* и свяжите его с атрибутом *TG*. Параметрами скрипта *TG Calculate*, вычисляющего градиент набора высоты при взлетном режиме, является атрибут *\_N\_dvig*. Напишите тело скрипта (рис. 2.38).

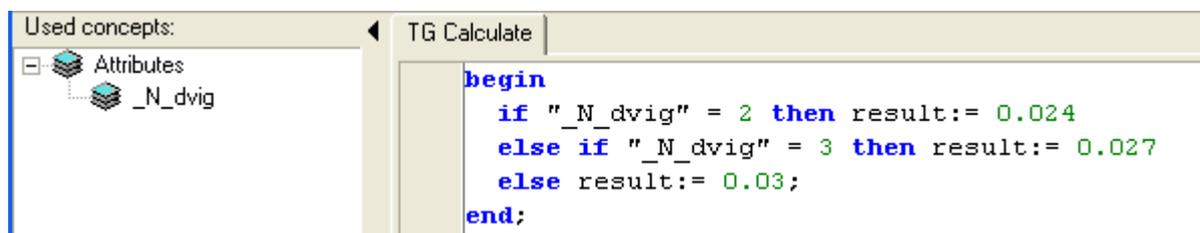


Рисунок 2.38 – Атрибуты и тело скрипта *TG Calculate*

- **Определение скрипта для вычисления тяговооруженности из условия набора высоты при одном отказавшем двигателе** (по формуле 2.16).

Созданный скрипт переименуйте в *PO\_3 Calculate* и свяжите его с атрибутом *PO\_3*. Параметрами скрипта *PO\_3 Calculate*, вычисляющего тяговооруженность из условия набора высоты при одном отказавшем двигателе, являются атрибуты *\_N\_dvig* и *TG*. Напишите тело скрипта (рис. 2.39).

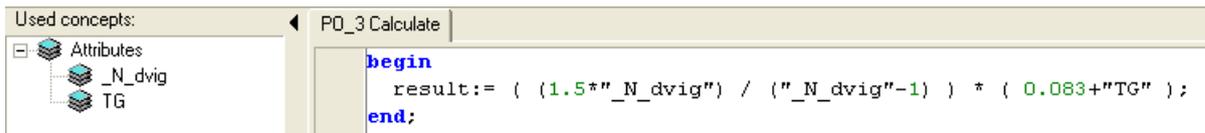


Рисунок 2.39 – Атрибуты и тело скрипта *PO\_3 Calculate*

- **Определение скрипта для вычисления потребной тяговооруженности** (по формуле 2.17).

Созданный скрипт переименуйте в *PO\_max Calculate* и свяжите его с атрибутом *PO\_max*. Параметрами скрипта *PO\_max Calculate*, вычисляющего потребную тяговооруженность, являются атрибуты *PO\_1*, *PO\_2*, *PO\_3* и вспомогательная переменная *Temp*. Напишите тело скрипта (рис. 2.40).

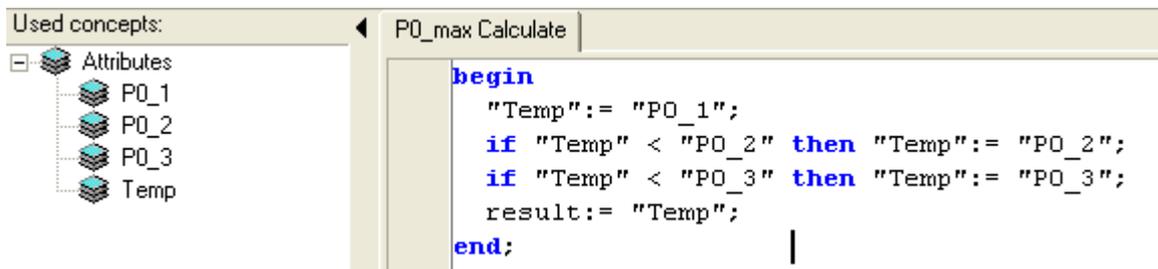


Рисунок 2.40 – Атрибуты и тело скрипта *PO\_max Calculate*

Таким образом, в дереве концептов дескриптивной онтологии предметной области “Выбор потребной тяговооруженности самолета” можно видеть следующие скрипты (рис. 2.41).

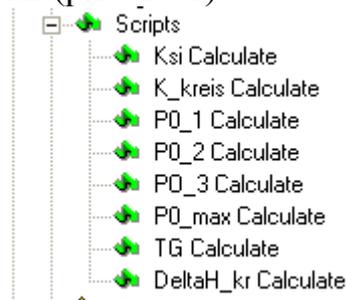


Рисунок 2.41 – Скрипты дескриптивной онтологии

#### 2.2.2.4. Представление онтологии в виде семантической сети

Дескриптивная онтология может быть представлена не только в виде дерева концептов, но также в виде семантической сети, представляющей собой ориентированный граф, в котором вершины представляют собой концепты онтологии, а ребра отображают связи между концептами.

Пользователь имеет возможность перемещать концепты семантической сети в пределах экрана, перетаскивая их с помощью мыши”.

Для того чтобы получить представление дескриптивной онтологии в виде семантической сети, необходимо выполнить следующую последовательность команд *Tools -> Ontology as network* .

В левой части открывшегося окна *Ontology Network* дескриптивная онтология отображается в виде дерева концептов, а в правой части – в виде семантической сети (рис. 2.42).

При закрытии окна *Ontology Network* происходит возврат в конструктор онтологий.

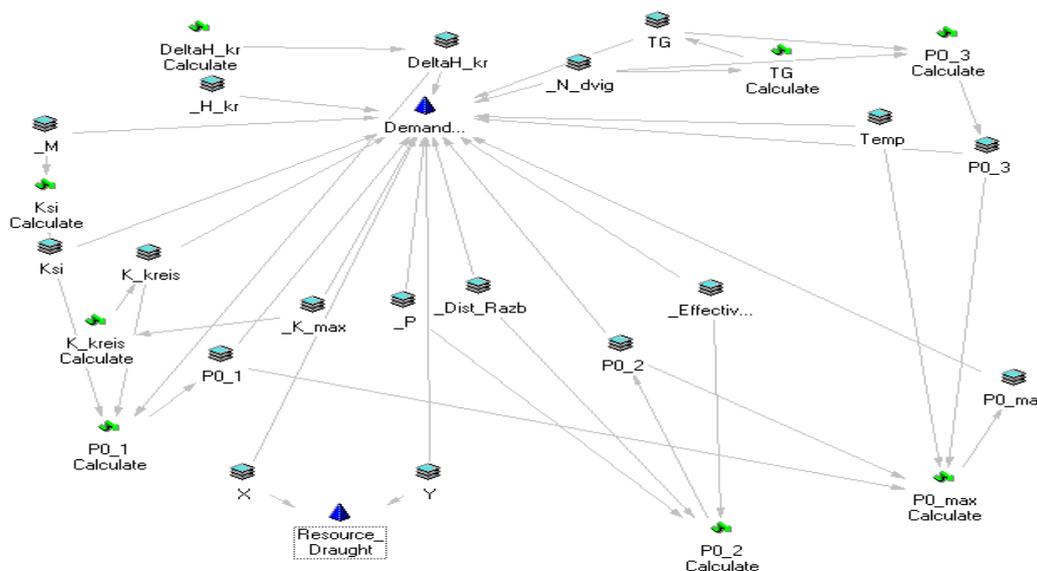


Рисунок 2.42 – Представление онтологии предметной области “Выбор потребной тяговооруженности самолета” в виде семантической сети

## 2.2.3. Проектирование онтологии мира заказов и ресурсов

### 2.2.3.1. Создание онтологии мира заказов и ресурсов

Для создания онтологии мира заказов и ресурсов (онтологии виртуального мира) необходимо выделить библиотеку онтологий и в контекстном меню выбрать пункт *New item -> Virtual world ontology*. При этом в правой части экрана появится диалог создания онтологии мира заказов и ресурсов, предоставляющий возможность выбора тех концептов «объект», для которых требуется создание агентов заказа или ресурса. Если предполагается, что какой-либо объект должен иметь одновременно и агента заказа, и агента ресурса, следует выставить флажок слева от названия этого объекта. Агенты будут созданы автоматически. Если объекту в виртуальном мире должен соответствовать либо агент заказа, либо агент ресурса, флажок выставлять не следует. Агенты будут созданы позже в индивидуальном порядке. Так, в данном примере объект “проектируемый самолет” выступает

в роли заказа и должен иметь в виртуальном мире только агента заказа. В свою очередь, объект-ресурс выступает в роли ресурса и должен иметь в виртуальном мире только агента ресурса. Нажатие на кнопку <OK> подтверждает необходимость создания онтологии виртуального мира (рис. 2.43).

При нажатии <OK> пиктограмма онтологии виртуального мира появляется в архитектуре онтологий в левой части экрана. При нажатии <+> раскрывается дерево концептов онтологии виртуального мира, которое содержит концепты агентов заказов и ресурсов, а также отношений между агентами (рис. 2.43).

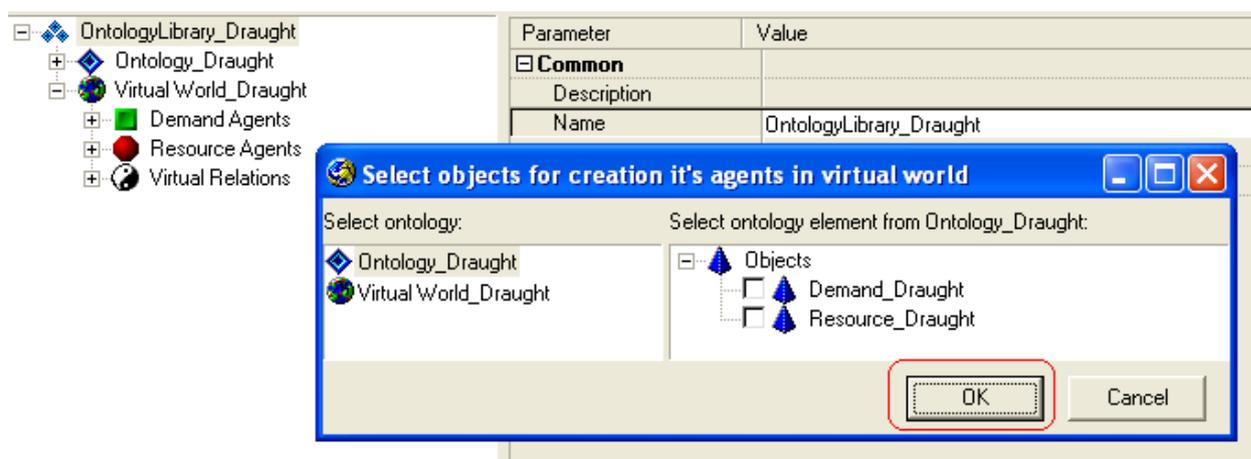


Рисунок 2.43 – Выбор объектов, для которых должны создаваться агенты

Создайте онтологию мира заказов и ресурсов для предметной области «Выбор удельной нагрузки на крыло» (*New Item -> Virtual World Ontology*). Введите в поле *Name* название онтологии - *Virtual World\_Draught*. Раскройте дерево концептов онтологии виртуального мира (рис. 2.44).

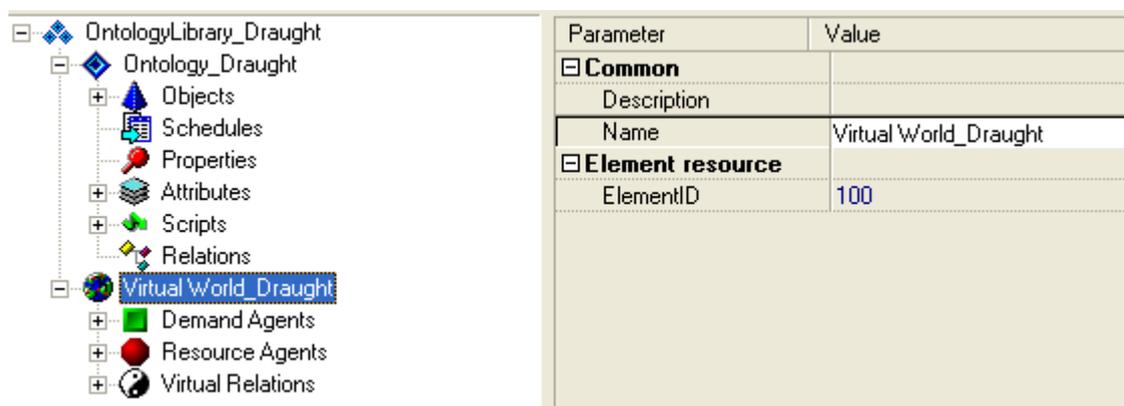


Рисунок 2.44 – Категории концептов онтологии мира заказов и ресурсов

### 2.2.3.2. Создание концепта «агент заказа»

Создайте концепт «агент заказа» для концепта «проектируемый самолет» (т. к. именно проект является активной сущностью): выделите категорию *Demand Agents*, в контекстном меню выберите пункт *New Item -> Demand agent* и в появившемся диалоге выберите концепт *Demand\_Draught*. Затем нажмите <OK> и переименуйте созданный концепт в *Demand\_Draught* (рис. 2.45). Назначьте концепту «агент заказа» три вида пиктограмм, с помощью которых данный концепт будет отображаться при работе со сценой в процессе моделирования. Установите у концепта флаг *vaoAutoCreate* (должен быть установлен по умолчанию).

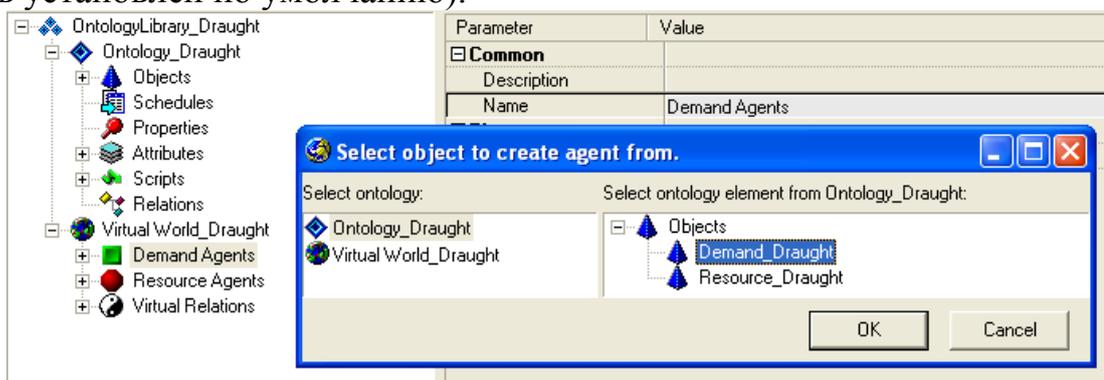


Рисунок 2.45 – Создание агента заказа для концепта *Demand\_Draught*

### 2.2.3.3. Создание концепта «агент ресурса»

Создайте концепт «агент ресурса» для концепта «объект-ресурс»: выделите категорию *Resource Agents*, в контекстном меню выберите пункт *New Item -> Resource agent* и в появившемся диалоге выберите концепт *Resource\_Draught*. Затем нажмите <OK> и переименуйте созданный концепт в *Resource\_Draught* (рис. 2.46). Назначьте концепту «агент ресурса» три типа пиктограмм, с помощью которых данный концепт будет отображаться при работе со сценой в процессе моделирования. Установите у концепта флаг *vaoAutoCreate*. Не устанавливайте у концепта флаг *raoActive*.

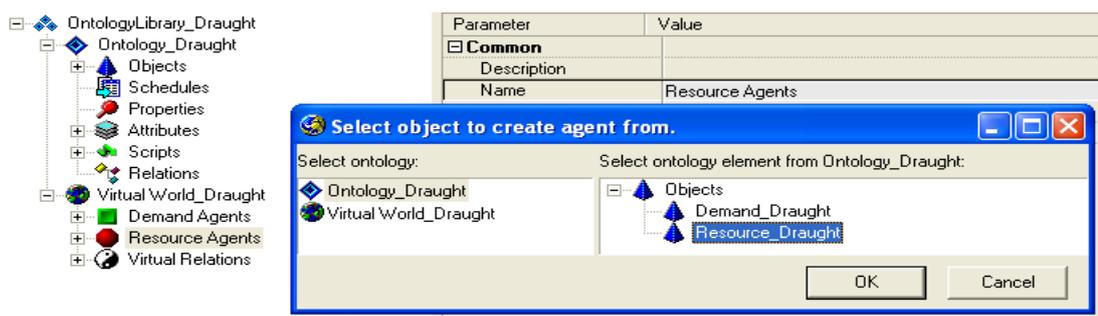


Рисунок 2.46 – Создание агента ресурса для концепта *Resource\_Draught*

#### 2.2.3.4. Виртуальные отношения: отношение матчинга

Отношение матчинга является служебным классом отношений в виртуальном мире и связывает между собой концепты заказов/ресурсов. Отношение матчинга показывает *возможность* матчинга между агентами, концепты которых в онтологии связаны данным отношением. Иными словами, матчинг возможен, но он не обязательно состоится: агенты могут не договориться по разным причинам (есть более выгодное предложение, данное предложение не устраивает партнера/агента и т.д.).

Отношение матчинга возможно только между агентами заказа и ресурса. Например, матчинг агента заказа с агентом другого заказа невозможен. Отношение матчинга является отношением вида «субъект-объект». Субъект выступает инициатором матчинга. Агент заказа и агент ресурса могут устанавливать отношение матчинга в сцене, причем инициатором (субъектом) матчинга может выступать как агент заказа, так и агент ресурса (если для него установлен флаг активности *raoActive*).

Установите отношение одностороннего матчинга между концептами агента заказа *Demand\_Draught* и агента ресурса *Resource\_Draught*. Для этого в категории *Virtual Relations* выделите концепт *Matching relation* и в контекстном меню выберите *Establish relation*. В правом окне раскройте дерево агентов виртуального мира, а в нем - категории *DemandAgents* и *ResourceAgents*. В качестве *Matching subject* выберите *Demand\_Draught*, а в качестве *Matching object* – *Resource\_Draught* (рис. 2.47).

Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Demand\_Draught.Resource\_Draught* (рис. 2.47).

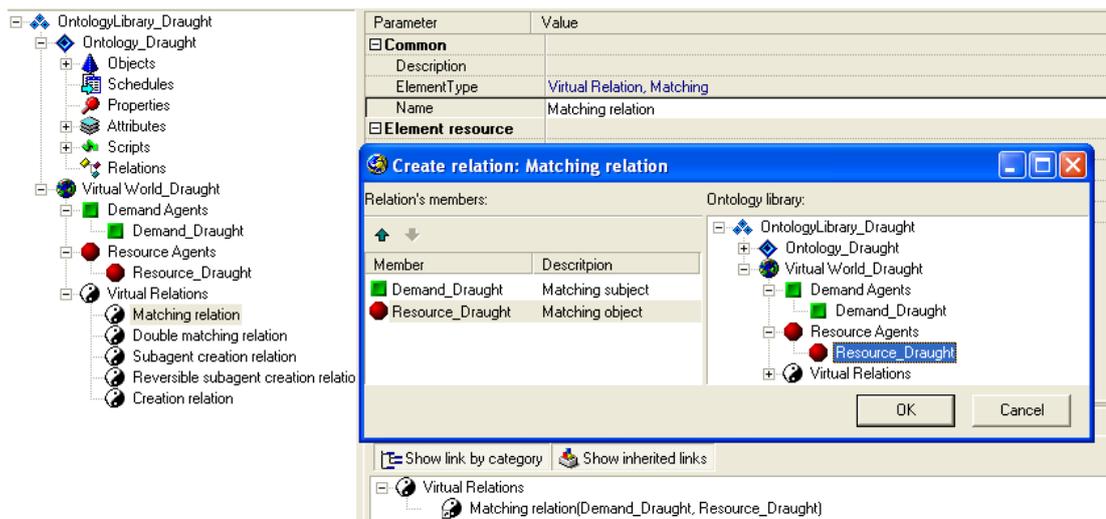


Рисунок 2.47 – Связывание отношением матчинга концептов *Demand\_Draught* и *Resource\_Draught*

### 2.2.3.5. Условия матчинга

Перейдите в закладку *Used by* (*Virtual relations -> Matching relation -> Used by*) и выделите *Matching relation (Demand\_Draught, Resource\_Draught)*, далее в контекстном меню выберите *Edit virtual relation properties* (рис. 2.48). При этом откроется окно редактирования условий матчинга (рис. 2.49).

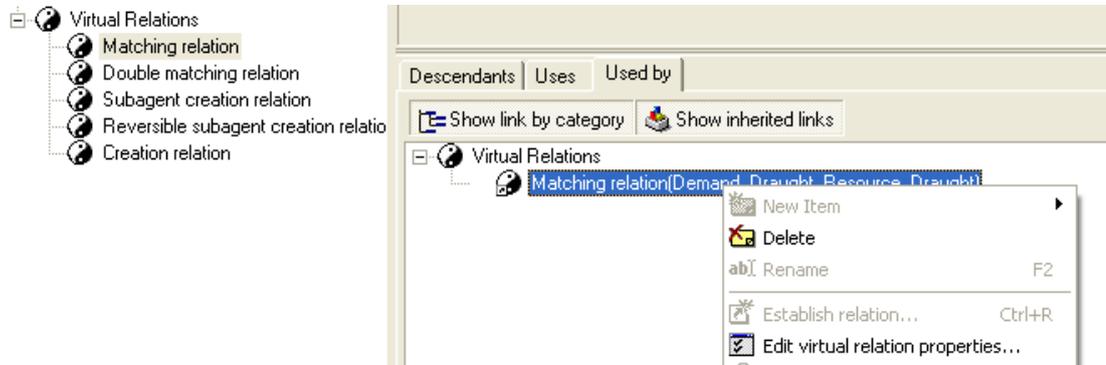


Рисунок 2.48 – Редактирование свойств отношения матчинга

В окне редактирования условий матчинга имеются следующие закладки:

- *Matching conditions* – создание и редактирование условий матчинга. Знаковое и скриптовое условия описаны выше. *Name* – тип условия (записывается автоматически), *Checking agent* – агент проверяющий условие матчинга, т.е. агент-субъект (записывается автоматически).
- *Decision Making Machine conditions* – создание и редактирование критериев, на основании которых в процессе матчинга принимается решение о резервировании агентом заказа агента ресурса.
- *Tasks* – формирование заданий на расчет дополнительных атрибутов, необходимых матчеру (в данной работе не используется).
- *Events* – обработчик событий, которые используются, когда необходимо изменить значение какого-либо атрибута агента в зависимости от значения атрибутов в матчере (в данной работе не используется).



Рисунок 2.49 – Окно редактирования условий матчинга

Так как в данной задаче матчинг необходим только для того, чтобы запустить скрипты, выполняющие расчеты, никаких ограничений не существует и никаких особых условий матчинга создавать не следует.

### 2.2.3.6. Условия принятия решения (*Decision Making Machine conditions*)

Условия принятия решения предназначены для работы машины принятия решений и позволяют агенту выбрать одно из множества возможных предложений (матчингов) от других партнёров. Условие задаётся в закладке *Decision Making Machine conditions* окна редактирования *Edit Matching Conditions* при помощи кнопки . Для условия принятия решения необходимо определить атрибут условия, направление оптимизации (максимум/минимум) и весовой коэффициент, определяющий «значимость» данного условия.

### 2.2.3.7. Создание условия принятия решения - выбор максимального значения тяговооруженности

За расчетное значение потребной тяговооруженности самолета согласно формуле (2.17) следует принять максимальное из трех значений: тяговооруженности из условия горизонтального полета на крейсерской скорости (формула (2.12)), тяговооруженности из условия заданной длины разбега (формула(2.15)) и тяговооруженности из условия набора высоты при одном отказавшем двигателе (формула (2.16)).

Создайте условие 1 принятия решений для матчинга *Wing\_Demand -> Wing\_Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия (рис. 2.50):

- Attribute = ‘Demand\_Draught.P0\_max’;
- Order = ‘Max’;
- Weight = ‘100’.

Активизируйте условие принятия решения, выставляя флажок *Active*. Условие принятия решения в матчинге для выполнения расчетов приведены на рис. 2.50.

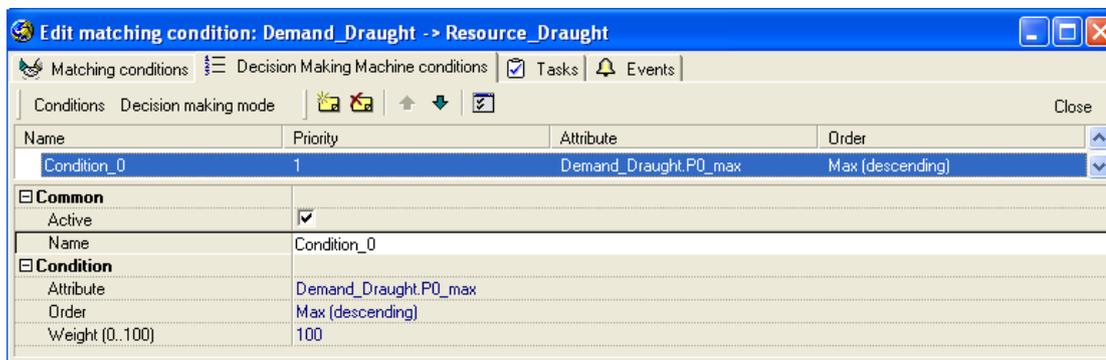


Рисунок 2.50 – Условие принятия решения в матчинге *Demand\_Draught - Resource\_Draught*

### 2.2.3.8. Параметры представления экземпляра объекта в сцене

Поведение экземпляра концепта «объект» (агента) в сцене определяет Группа свойств концепта «объект» *Interface behaviour* (рис. 2.51).

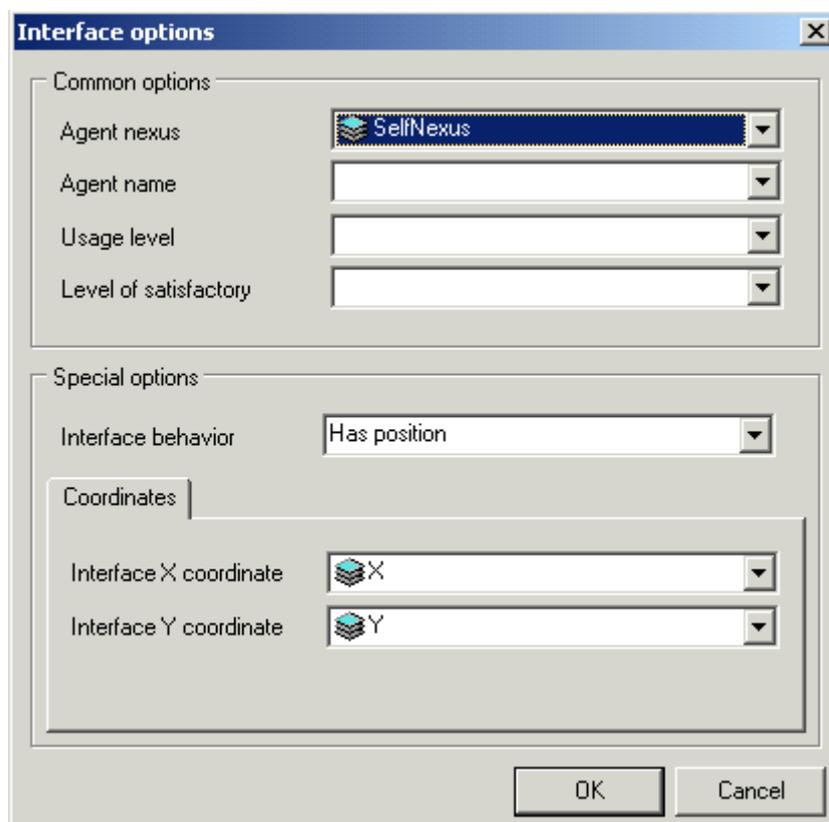


Рисунок 2.51 – Редактор свойств *Interface behaviour*

Свяжите атрибуты *X*, *Y* с позицией агента объекта *Demand\_Draught* в сцене. Для этого в редакторе свойств объекта *Demand\_Draught* вызовите диалог редактирования свойства *Interface behaviour*, в списке атрибутов этого свойства выберите параметр *Has position*. Затем в качестве *Interface X coordinate* и *Interface Y coordinate* выберите атрибуты *X* и *Y* соответственно. Нажмите <OK>. Свяжите атрибуты *X*, *Y* с позицией агента объекта *Resource\_Draught* в сцене. Для этого в редакторе свойств объекта *Resource\_Draught* вызовите диалог редактирования свойства *Interface behaviour*, в списке атрибутов этого свойства выберите параметр *Has position*. Затем в качестве *Interface X coordinate* и *Interface Y coordinate* выберите атрибуты *X* и *Y* соответственно. Нажмите <OK>.

### 2.2.3.9. Сохранение онтологий предметной области “Выбор потребной тяговооруженности самолета”

Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор потребной тяговооруженности самолета” приведено на рис. 2.52.

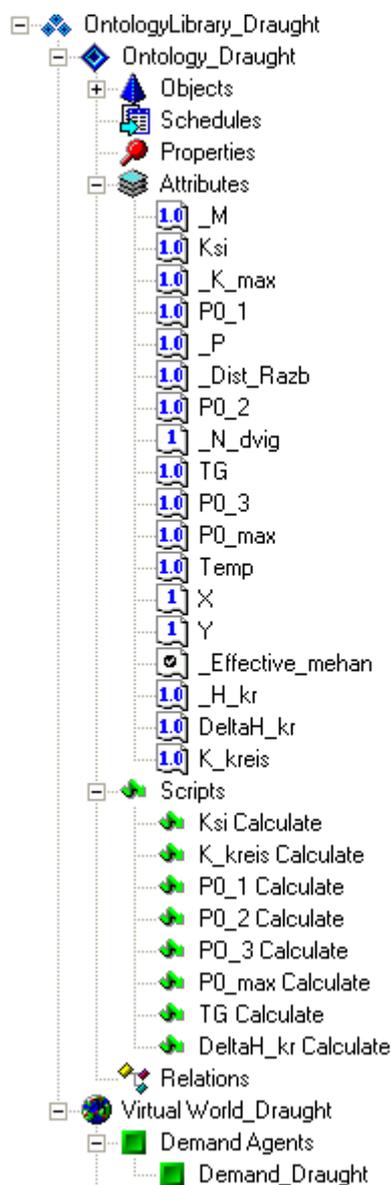


Рисунок 2.52 – Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор потребной тяговооруженности самолета”

Сохраните созданные онтологии (дескриптивную онтологию и онтологию мира заказов/ресурсов) при помощи кнопки  под именем *Ontology\_Draught*. Расширение *.ocl* будет добавлено автоматически. По умолчанию, файл онтологии будет размещен в разделе *Ontology Samples*. Завершите работу с конструктором онтологий (*File -> Close*).

#### 2.2.4. Создание онтологической сцены

Выполните на Вашем компьютере следующие настройки: *Пуск -> Панель управления -> Язык и региональные стандарты -> Настройка -> в поле “Разделитель целой и дробной части” установите точку* (рис. 2.53).

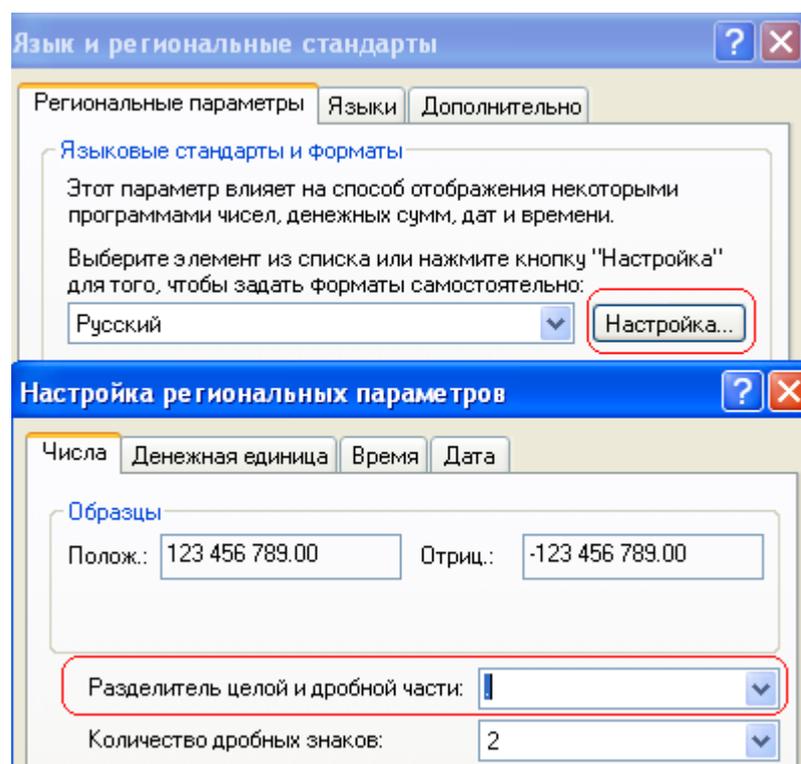


Рисунок 2.53 – Настройки компьютера

Вызовите исполняющую систему, запуская программу  `Unifntf.exe`, находящуюся в папке `OntConsUnifntf`. Создайте новую онтологическую сцену (`File` → `New scene` → `Load ontology`, выберите онтологию `Ontology_Draught.ocl`). В окне физического мира создайте одного агента проектируемого самолета `Demand_Draught_1` и одного агента объекта-ресурса `Resource_Draught_1` для выполнения расчетов (рис. 2.54). С помощью инспектора агентов установите следующие значения атрибутов для агентов, например, для самолета Ту-154.

<i>Атрибуты</i>	<i>Ту-154</i>	<i>Ил-86</i>	<i>Як-40</i>
<code>_Effective_mehan</code>	+	-	+
<code>_P</code>	<b>583,3</b>	<b>531,1</b>	<b>301,7</b>
<code>_Dist_razb</code>	1400	2000	750
<code>_N_dvig</code>	3	4	3
<code>_K_max</code>	<b>16,84</b>	<b>16,47</b>	<b>17,72</b>
<code>_M</code>	0,88	0,88	0,47
<code>_H_kr</code>	10100	10000	8100

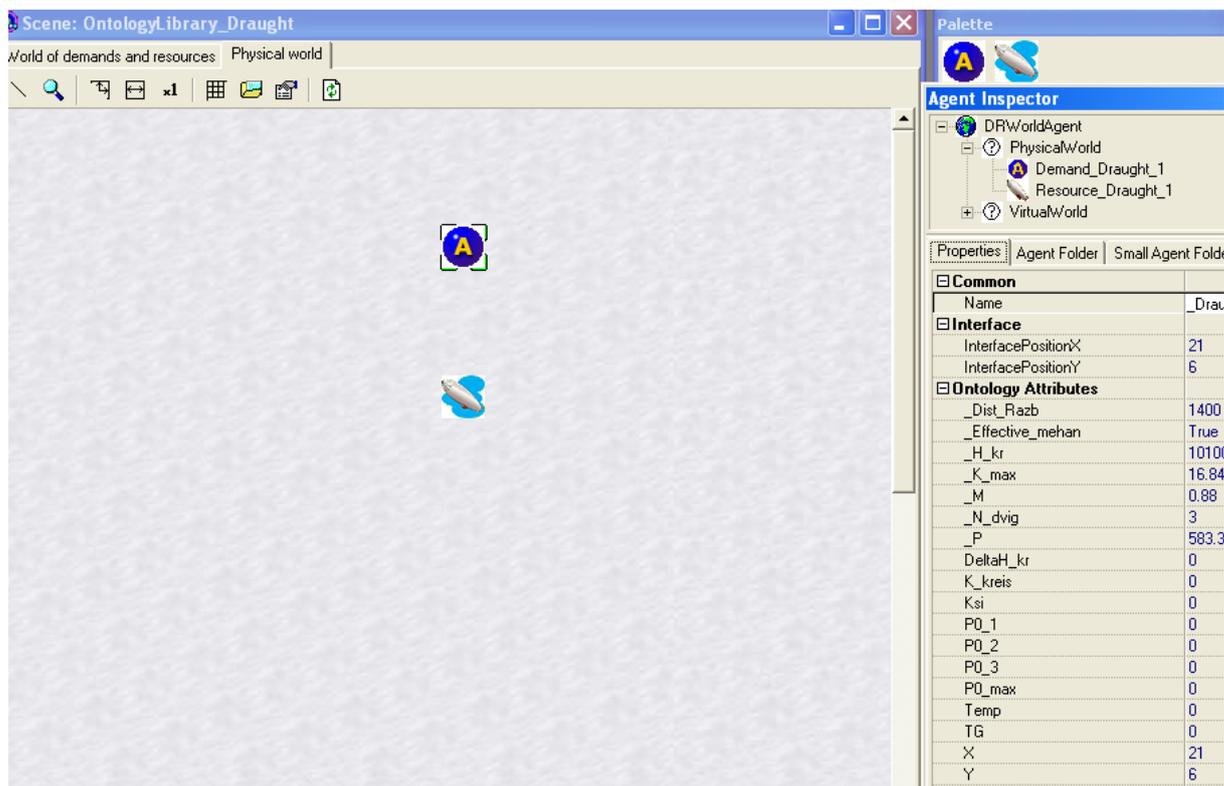


Рисунок 2.54 – Создание онтологической сцены предметной области «Выбор потребной тяговооруженности самолета»

## 2.2.5. Моделирование сцены виртуального мира

### 2.2.5.1. Запуск сцены на моделирование для самолета Ту-154

Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*). При помощи кнопки ►► запустите сцену на моделирование (т.е., запустите выполнение процесса матчинга). Наблюдайте процесс матчинга между агентами проектируемого самолета и агентами самолетов-прототипов в базе данных.

В процессе матчинга активный агент проектируемого самолета запускает скрипты, необходимые для вычисления значений атрибутов. Далее агент проекта строит таблицу принятия решений, в которой указан единственный зарезервированный агент объекта-ресурса. По двойному клику левой кнопкой мыши на пиктограмме агента *Demand\_Draught\_1* в окне физического или виртуального мира (либо при выполнении в контекстном меню агента команд *Actions -> Show agent structure*) можно открыть таблицу, в которой представлена структура агента (в закладке *Matchers*). В таблице указывается список ресурсов (партнеров), т.е. агентов, которые зарезервированы агентом-проектом (рис.2.55). При выделении партнера в правом окне отображаются атрибуты агента-проекта (простые (*Simple*), значения которых задавались непосредственно в сцене, и вычисляемые (*Scripted*), значения которых рассчитывались в процессе матчинга с помощью скриптов).

Partner	Status	Name	Value	State	Type
Resource_Draught_1	Accept	P0_max	0.309738	Assigned	Scripted
		Temp	0	Assigned	Simple
		P0_3	0.2475	Assigned	Scripted
		TG	0.027	Assigned	Scripted
		_N_dvig	3	Assigned	Simple
		P0_2	0.309738	Assigned	Scripted
		_Effective_mehan	True	Assigned	Simple
		_Dist_Razb	1400	Assigned	Simple
		_P	583.3	Assigned	Simple
		P0_1	0.216395295275356	Assigned	Scripted
		DeltaH_kr	0.3354	Assigned	Scripted
		_H_kr	10100	Assigned	Simple
		K_kreis	13.472	Assigned	Scripted
		_K_max	16.84	Assigned	Simple
		Ksi	1.02134528	Assigned	Scripted
		_M	0.88	Assigned	Simple

Рисунок 2.55 – Структура агента проекта *Demand\_Draught\_1* ТУ-154

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 2.56), в которой перечислены основные параметры, вычисляемые с помощью скриптов).

Agent	P0_max
Resource_Draught_1	0.309738

Рисунок 2.56 – Таблица принятия решений агента *Demand\_Draught\_1* ТУ-154

Агент-ресурс в данной задаче не имеет атрибутов, поэтому таблица, содержащая его структуру, пуста.

Окончательные результаты матчинга можно видеть на рисунке 2.57. В результате матчинга была выполнена следующая операция резервирования:  
– *Demand\_Draught\_1* – *Resource\_Draught\_1*.



Рисунок 2.57 – Результаты матчинга

### 2.2.5.2. Сохранение сцены виртуального мира

Для сохранения сцены используются средства онтологического сохранения: кнопка  или последовательность команд *Tools - > Save*

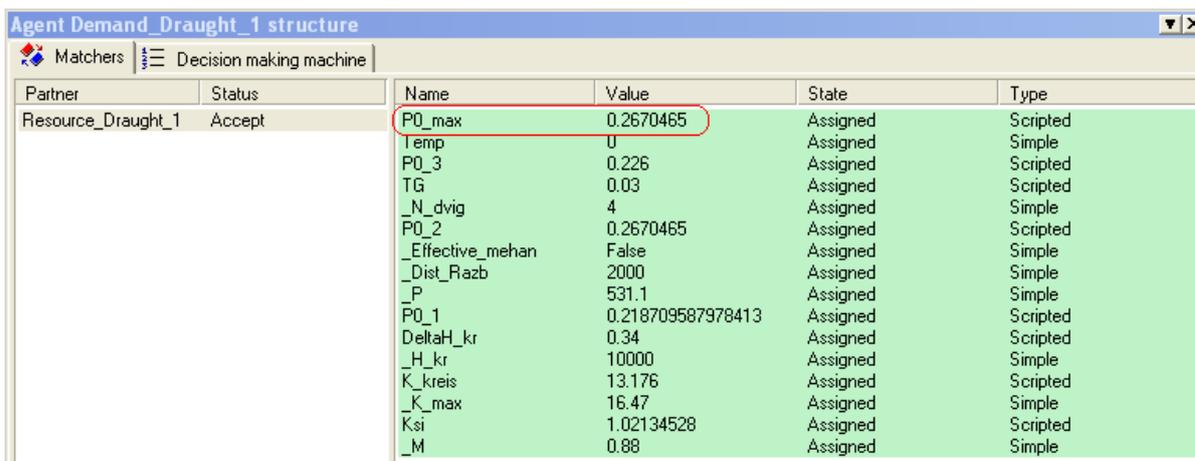
*ontology scene...* Сохраните сцену под именем *Scene\_Draught\_TY-154*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*. Завершите работу с исполняющей системой (*File -> Close*).

### 2.2.5.3. Загрузка ранее созданной сцены виртуального мира

Для загрузки ранее созданной сцены используются средства онтологической загрузки: кнопка  или последовательность команд *Tools -> Load ontology scene...* Загрузите сцену *Scene\_Draught\_TY-154.osf*.

### 2.2.5.4. Моделирование сцены для самолета Ил-86

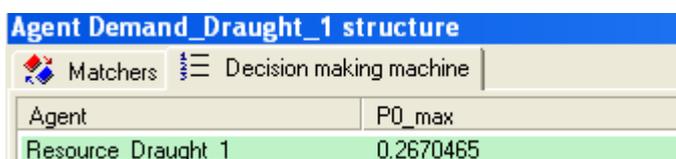
Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*). У агента проекта *Demand\_Draught\_1* измените значения атрибутов для самолета Ил-86. Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*). При помощи кнопки  запустите сцену на моделирование (т.е., запустите процесс матчинга). Наблюдайте процесс матчинга между агентами проекта и самолетов-прототипов. Результаты матчинга показаны на рис.2.58, а таблица принятия решения агента проекта – на рис. 2.59.



Partner	Status	Name	Value	State	Type
Resource_Draught_1	Accept	P0_max	0.2670465	Assigned	Scripted
		Temp	0	Assigned	Simple
		P0_3	0.226	Assigned	Scripted
		TG	0.03	Assigned	Scripted
		_N_dvig	4	Assigned	Simple
		P0_2	0.2670465	Assigned	Scripted
		_Effective_mehan	False	Assigned	Simple
		_Dist_Razb	2000	Assigned	Simple
		_P	531.1	Assigned	Simple
		P0_1	0.218709587978413	Assigned	Scripted
		DeltaH_kr	0.34	Assigned	Scripted
		_H_kr	10000	Assigned	Simple
		K_kreis	13.176	Assigned	Scripted
		_K_max	16.47	Assigned	Simple
		Ksi	1.02134528	Assigned	Scripted
		_M	0.88	Assigned	Simple

Рисунок 2.58 – Структура агента проекта *Demand\_Draught\_1* Ил-86

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис.2.59), в которой перечислены основные параметры, вычисляемые с помощью скриптов).



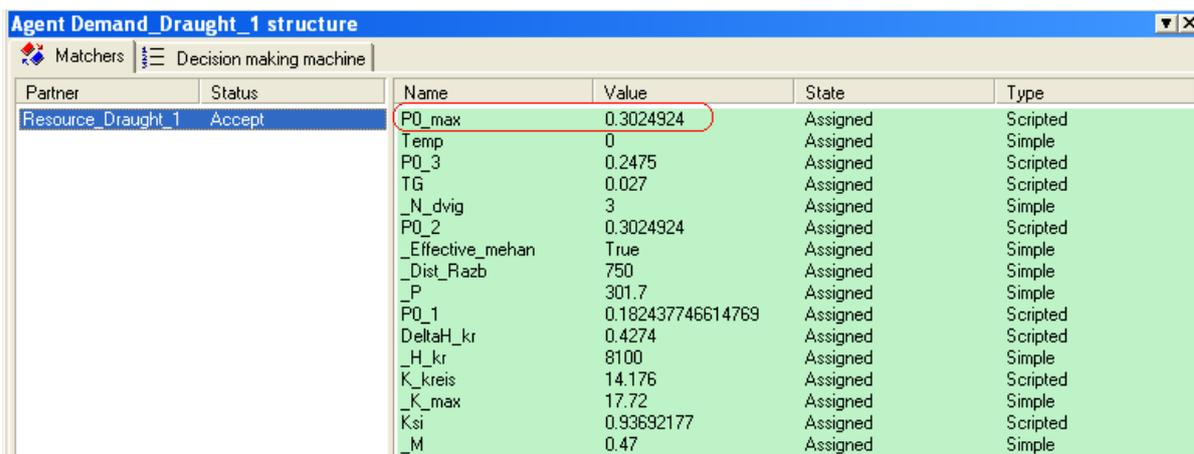
Agent	P0_max
Resource_Draught_1	0.2670465

Рисунок 2.59 – Таблица принятия решений агента *Demand\_Draught\_1* Ил-86

Агент-ресурс в данной задаче не имеет атрибутов, поэтому таблица, содержащая его структуру, пуста. При помощи средств онтологического сохранения сохраните сцену под именем *Scene\_Draught\_ИЛ-86*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*. Завершите работу с исполняющей системой (*File -> Close*).

### 2.2.5.5. Моделирование сцены для самолета Як-40

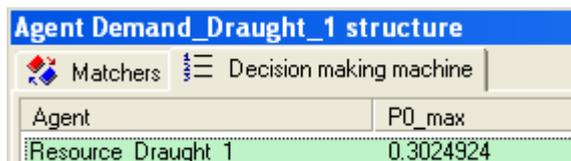
При помощи средств онтологической загрузки загрузите сцену *Scene\_Draught\_TY-154.osf*. Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*). У агента проекта *Demand\_Draught\_1* измените значения атрибутов для самолета Як-40. Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*). При помощи кнопки  запустите сцену на моделирование (т.е., запустите процесс матчинга). Наблюдайте процесс матчинга между агентами проекта и самолетов-прототипов. Результаты матчинга показаны на рис. 2.60, а таблица принятия решения агента проекта – на рис. 2.61.



Partner	Status	Name	Value	State	Type
Resource_Draught_1	Accept	P0_max	0.3024924	Assigned	Scripted
		Temp	0	Assigned	Simple
		P0_3	0.2475	Assigned	Scripted
		TG	0.027	Assigned	Scripted
		_N_dvig	3	Assigned	Simple
		P0_2	0.3024924	Assigned	Scripted
		_Effective_mehan	True	Assigned	Simple
		_Dist_Razb	750	Assigned	Simple
		_P	301.7	Assigned	Simple
		P0_1	0.182437746614769	Assigned	Scripted
		DeltaH_kr	0.4274	Assigned	Scripted
		_H_kr	8100	Assigned	Simple
		K_kreis	14.176	Assigned	Scripted
		_K_max	17.72	Assigned	Simple
		Ksi	0.93692177	Assigned	Scripted
		_M	0.47	Assigned	Simple

Рисунок 2.60 – Структура агента проекта *Demand\_Draught\_1* Як-40

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 30), в которой перечислены основные параметры, вычисляемые с помощью скриптов).



Agent	P0_max
Resource_Draught_1	0.3024924

Рисунок 2.61 – Таблица принятия решений агента *Demand\_Draught\_1* Як-40

Агент-ресурс в данной задаче не имеет атрибутов, поэтому таблица, содержащая его структуру, пуста.

При помощи средств онтологического сохранения сохраните сцену под именем *Scene\_Draught\_ЯК-40*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*. Завершите работу с исполняющей системой (*File -> Close*).

### 2.2.6. Контрольные вопросы

1. Дайте определения следующих понятий: онтология, дескриптивная онтология, онтология мира заказов и ресурсов, концепт, категория концептов.
2. Каковы основные функции конструктора онтологий?
3. Опишите структуру библиотеки онтологий. Какие основные категории концептов в ней присутствуют?
4. Для чего предназначено дерево категорий и концептов менеджера конструктора онтологий?
5. Как создать концепт дескриптивной онтологии и задать его свойства (на примере концептов «объект» и «атрибут»)? Как связать концепты между собой?
6. Как определить скрипт для вычисления значения атрибута концепта? Какие функции выполняет редактор скриптов?
7. Как проверить синтаксис скрипта? Как сохранить скрипт?
8. Как создать концепт онтологии мира заказов/ресурсов и задать его свойства (на примере концептов «заказ» и «ресурс»)?
9. Как установить отношение матчинга между концептами мира заказов/ресурсов?
10. Как сконструировать онтологическую сцену с помощью инструментов, предоставляемых исполняющей системой?
11. Как проанализировать результаты, полученные в результате выполнения скриптов? Рассмотрите структуру агентов.
12. Как установить или изменить значения атрибутов агентов с помощью Инспектора агентов?
13. Какие действия необходимо выполнить в процессе моделирования онтологической сцены?

### 3. ЛАБОРАТОРНАЯ РАБОТА №3

#### ВВЕДЕНИЕ

В качестве примера выполнения онтологического анализа с помощью редактора онтологий **Protégé 4.1 beta** рассмотрена предметная область «самолет». Версия 4.1. Protégé позволяет построить онтологию для семантической сети на OWL (Web Ontology Language). Формальная семантика OWL дает возможность получения логических следствий, то есть фактов, которые не присутствуют непосредственно в онтологии, но могут быть выведены из существующих посредством семантики.

**OWL** (англ. *Web Ontology Language*) — язык описания онтологий для семантической паутины. Язык OWL позволяет описывать классы и отношения между ними, присущие для веб-документов и приложений. OWL основан на более ранних языках OIL и DAML+OIL и в настоящее время является рекомендованным консорциумом Всемирной паутины.

В основе языка — представление действительности в модели данных «объект — свойство». OWL используется для описания не только веб-страниц, но и любых объектов действительности. Каждому элементу описания в этом языке (в том числе свойствам, связывающим объекты) ставится в соответствие свой **URI** (Uniform Resource Identifier) — унифицированный (единообразный) идентификатор ресурса. Прежде всего, речь идёт, конечно, о ресурсах сети [Интернет](#) и [Всемирной паутины](#). URI предоставляет простой и расширяемый способ идентификации ресурсов.

#### 3.1. ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ №3

##### Цели лабораторной работы:

1. Освоить на практике онтологический анализ предметной области «САМОЛЕТ» с помощью редактора онтологий Protege 4.1 beta ,
2. Закрепить навыки создания онтологии.

##### Задачи работы:

1. Создать фрагмент онтологии самолета.
2. Расширить онтологию в соответствии с полученным вариантом задания.
3. Наполнить онтологию значениями экземпляров.
4. Представить результат в виде семантической сети.

## 3.2. СОЗДАНИЕ ОНТОЛОГИИ

### 3.2.1. Создание проекта

Работа в системе Protege 4.1 beta, как и в Protege 3.3, начинается с создания нового проекта или выбора существующего.

Для создания нового проекта после запуска программы в окне приветствия (Рисунок 3.1) выберите кнопку «Create new OWL ontology».



Рисунок 3.1 – Окно приветствия

В появившемся окне "Create ontology wizard" (Рисунок 3.2) нажмите кнопку «Continue» два раза подряд.

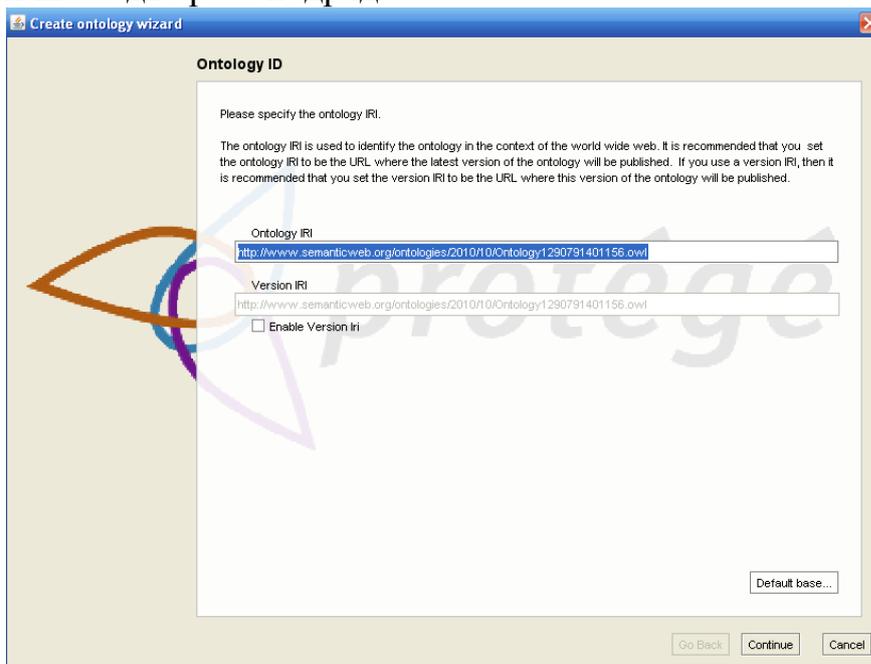


Рисунок 3.2 – Окно "Create ontology wizard"

После этого в появившемся окне (Рисунок 3.3) выберете формат OWL/XML и нажмите кнопку «Finish».

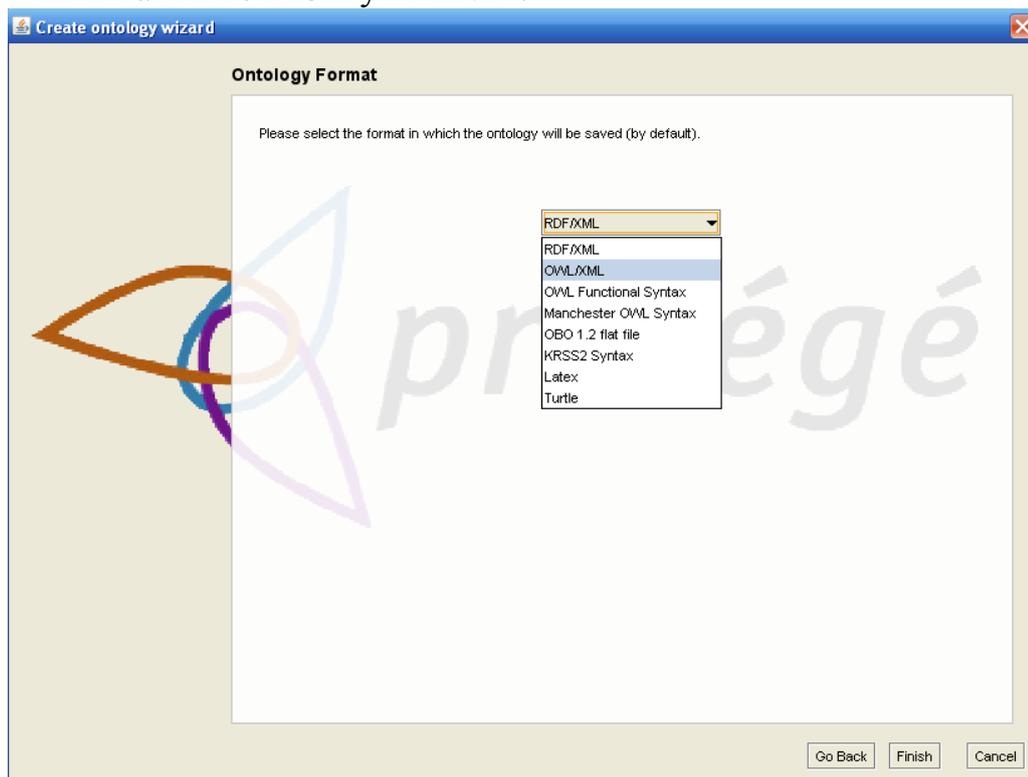


Рисунок 3.3 – Окно выбора формата онтологии

Далее откроется окно проекта Protégé (Рисунок 3.4).

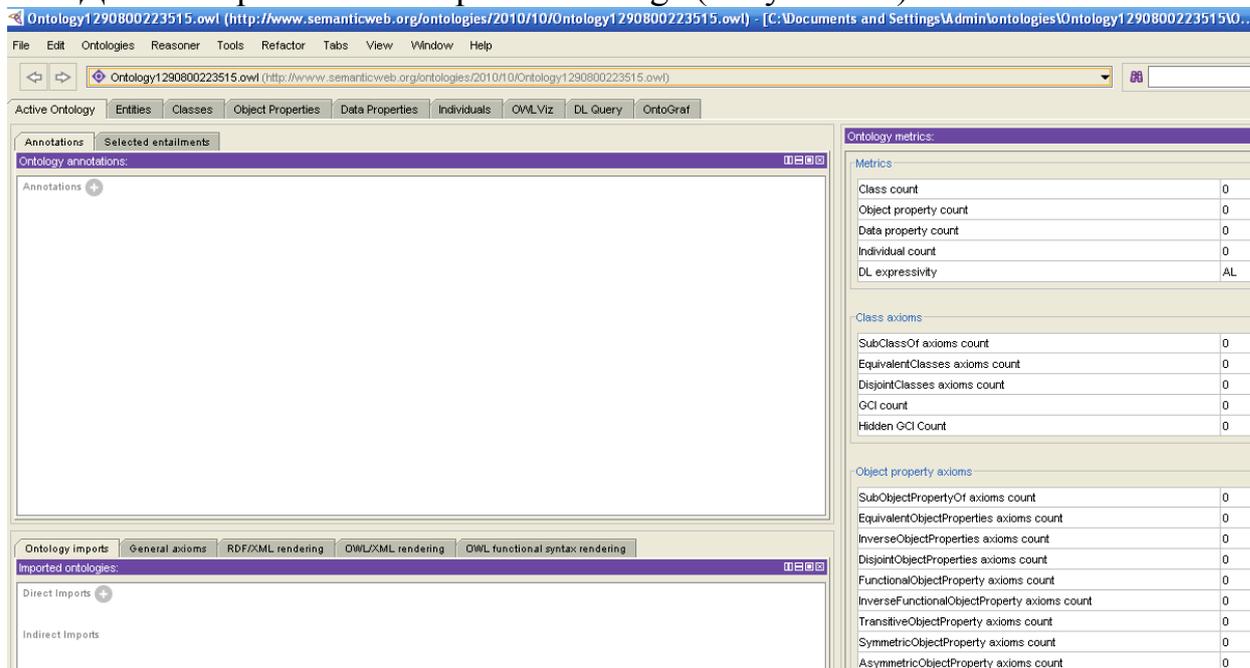


Рисунок 3.4 – Окно проекта

### 3.2.2. Сохранение проекта

Сохранить проект можно несколькими способами:

1. выбрав из меню File пункт «Save as...» (Рисунок 3.5),

2. нажав на панели инструментов кнопку сохранения проекта,
3. при последовательном нажатии клавиш Ctrl+Shift+S.

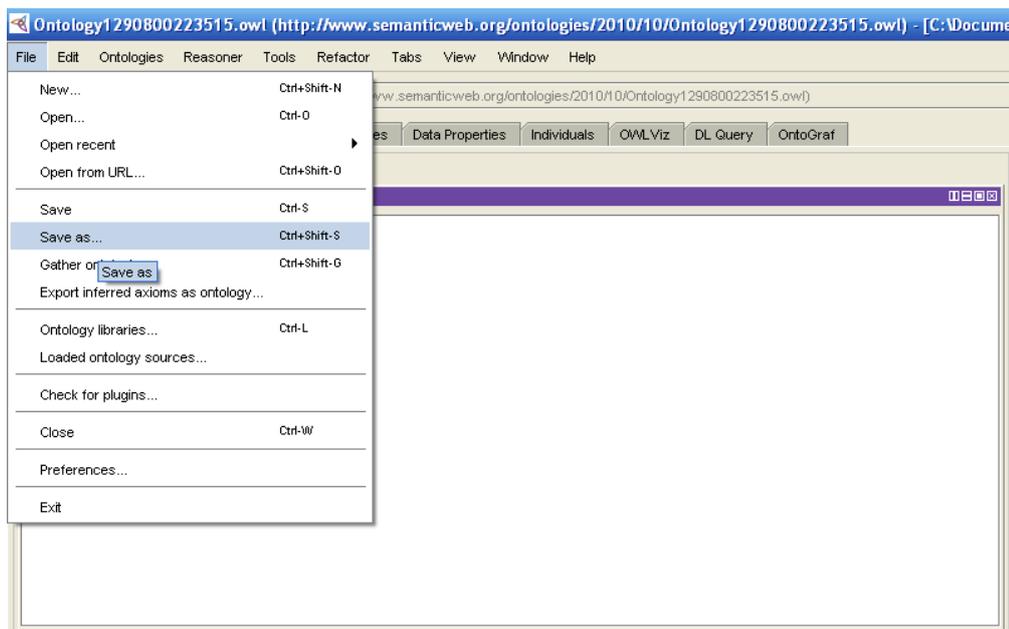


Рисунок 3.5 – Окно сохранения проекта

В появившемся окне выбора формата (Рисунок 3.6) нажмите ОК.

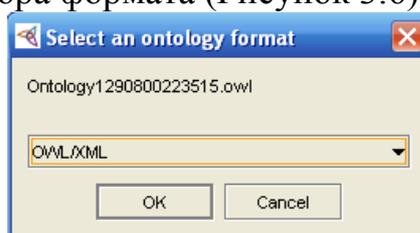


Рисунок 3.6 – Окно выбора формата

После этого появится окно выбора имени файла проекта (Рисунок 3.7). Измените стандартное имя на «LR3\_varX», где X номер Вашего варианта, выберите место для сохранения и нажмите ОК.

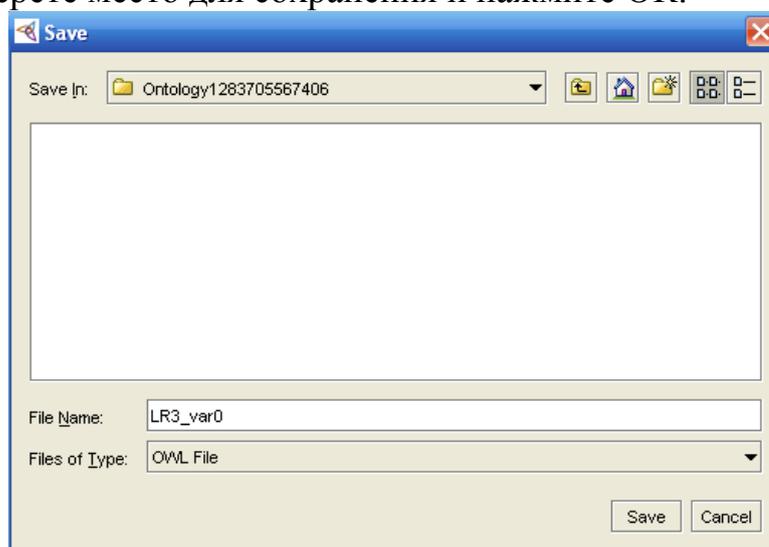


Рисунок 3.7 – Окно ввода имени файла

### 3.2.3. Создание классов

Основное окно программы Protégé (Рисунок 3.8) состоит из закладок, которые отображают различные аспекты модели знаний. При создании нового проекта наиболее важной закладкой является закладка классов (Classes). Классы соответствуют объектам или типам объектов, в некоторой предметной области. В лабораторной работе, будет создан один класс «Самолет», остальные классы будут являться его подклассами.

Классы в Protégé отображаются в виде иерархии наследования (Class hierarchy), которая располагается в левой части закладки классов. Информация о классе, выбранном в текущий момент в навигаторе, будет отображена в редакторе классов справа в полях Annotations, Usage.

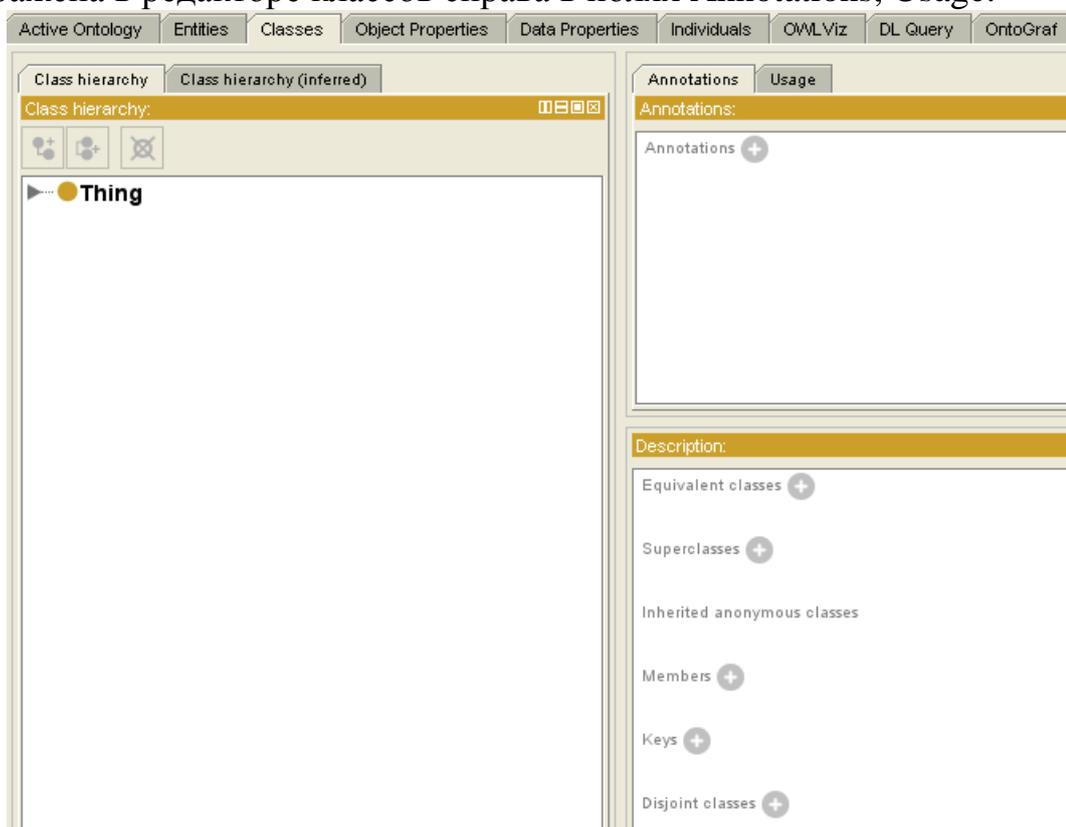


Рисунок 3.8 – Основное окно программы

### 3.2.4. Создание класса «Самолет»

Для того чтобы создать новый класс в навигаторе классов (Class Browser), выделите класс Thing (вещь, нечто). Все создаваемые классы должны находиться на уровень ниже этого системного класса.

Нажмите кнопку  добавить subclass (Add subclass) в верхнем правом углу навигатора классов. В появившемся окне «Create a new OWL Class» введите название класса «Самолет» и нажмите ОК (Рисунок 3.9).

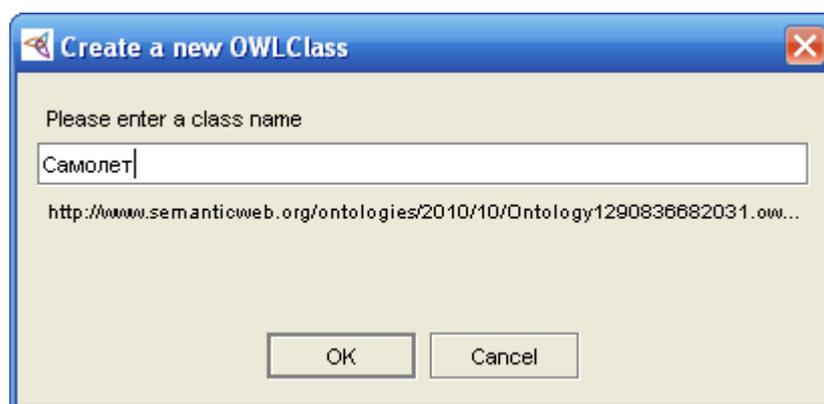


Рисунок 3.9 – Окно ввода имени класса

В окне навигатора классов появится созданный класс. Вы можете увидеть, что имя нового класса в навигаторе классов после создания будет выделено, для указания того, что этот класс выбран в данный момент (Рисунок 3.9).

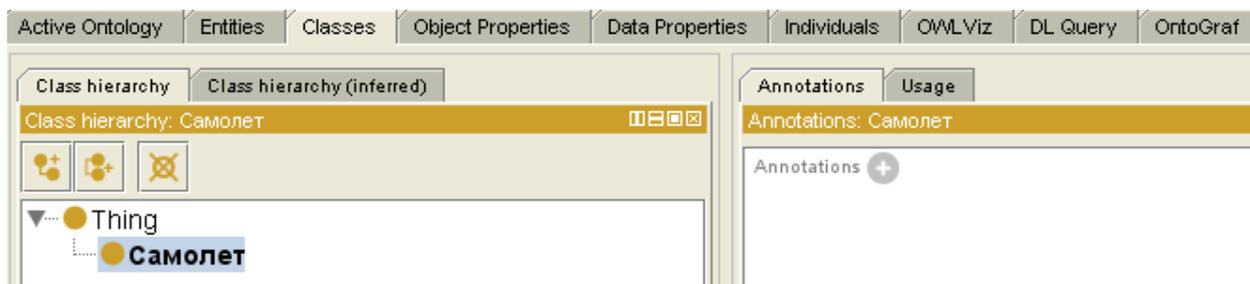


Рисунок 3.9 – Созданный класс «Самолет»

### 3.2.5. Создание подклассов класса «Самолет»

Создание подклассов аналогично созданию классов. Обратите внимание на различие между кнопками  – Add subclass (создание класса на уровень ниже выбранного) и  – Add sibling class (создание класса на том же уровне, что и выбранный). Кнопка  служит для удаления классов.

Для создания подкласса выделите класс «Самолет». Нажмите кнопку  добавить subclass (Add subclass) в верхнем правом углу навигатора классов. В появившемся окне «Create a new OWL Class» введите название класса и нажмите ОК.

Создайте фрагмент онтологии самолета [4, с.49-50]. Обратите внимание, что каждый элемент является классом в онтологии. Классы, находящиеся на уровень ниже класса «САМОЛЕТ» выделены, остальные классы разбиты с помощью скобок. Элемент перед скобками – класс, находящийся на уровень выше элементов в скобках. Будьте внимательны, соблюдая иерархию наследования. Словосочетание «и так далее» означает, что приведена лишь часть составляющих класса, и онтологию можно расширять.

## **САМОЛЕТ**

***(экипаж,***

***целевая нагрузка***

*(вид* (пассажиры, груз, вооружение, оборудование и т.д.),

*характер* (сбрасываемая, сгружаемая, несъемная),

*параметры* (количество, масса, расположение и т.д.),

*условия транспортировки*

*(комфорт* (количество стюардесс, количество буфетов, объем пассажирской кабины, расстояние между креслами, ширина прохода и т.д.),  
*безопасность и т.д.),*

*расположение*

*(снаружи* (*съемные gondoly* (на фюзеляже, на крыле)),

*внутри* (в фюзеляже, в крыле)),

*крепление* (грузовой пол, кресла, узлы подвески),

*загрузка\_выгрузка*

*(средства*

*(аэродромная механизация* (лестница самоходная, транспортер, кран),

*борт\_механизация* (лестница откидная, трап, лебедка),

*аварийные средства* (парашют, катапульта),

*характер* (время загрузки, время выгрузки),

*места*

*(двери пассажирские* (количество, размер, расположение),

*двери грузовые, люки),*

*и т.д.),*

***главные несущие поверхности,***

***энергетическая установка,***

***система стабилизаторов управления,***

***взлетно-посадочные устройства,***

***связующая конструкция и т.д.),***

При создании онтологии приведенного фрагмента в дереве иерархии расширяется только один класс – «Целевая нагрузка», он содержит семь подклассов, которые в свою очередь содержат от двух до четырех классов, а также имеет в дереве иерархии не менее трех уровней. Подклассы для других классов не показаны. При выполнении индивидуального задания необходимо аналогичным образом расширить онтологию, начиная с класса, соответствующего варианту задания (см. таблица 3.1), руководствуясь примерным фрагментом [4] и учебником [5]. Созданный фрагмент описанной онтологии самолет приведен на рисунке 3.10.

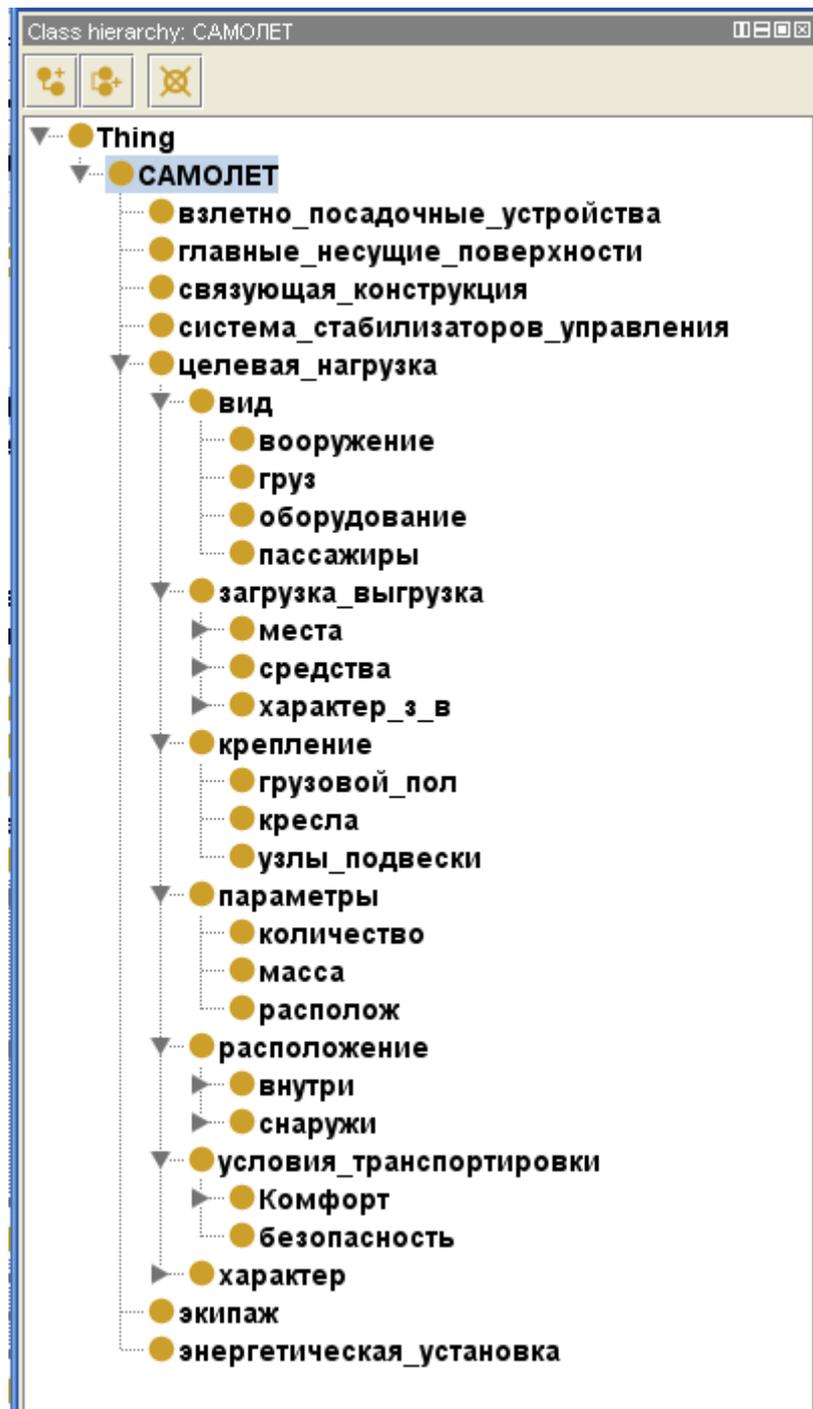


Рисунок 3.10 – Фрагмент онтологии самолета

### 3.2.6. Создание суперклассов

Создание суперклассов необходимо в случае повторения подклассов для нескольких классов. Класс «Размер» является универсальным параметром, он может применяться для определения габаритов (длина, ширина, высота и др., см. п.3.2.7.) различных составляющих самолета, например *размер двери* или *размер люка*. В рассматриваемом примере, этот подкласс должен содержаться в классах «Двери грузовые», «Двери пассажирские» и «Люки».

После создания подкласса «Размер» в одном из указанных выше классов, например «Двери грузовые» (Рисунок 3.11), создать подкласс с таким же именем в других классах, нельзя.

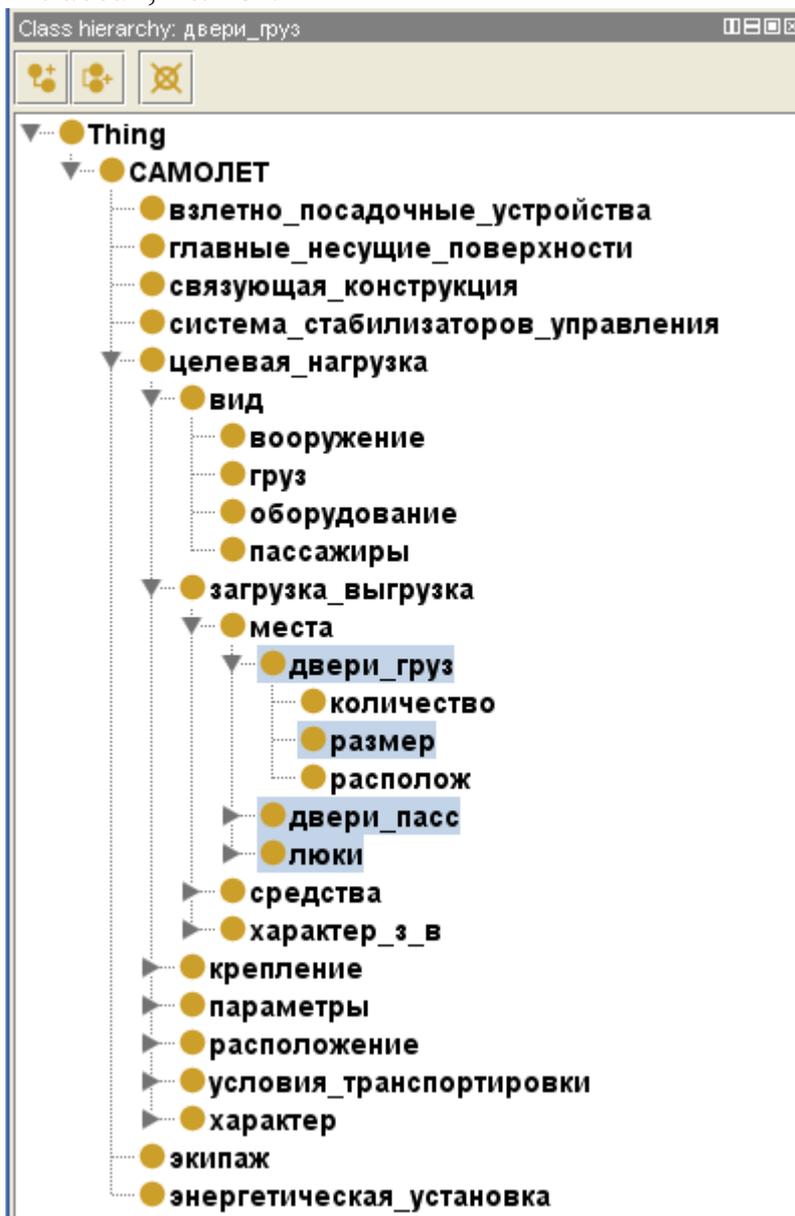


Рисунок 3.11 – Классы «Двери грузовые» содержащий подкласс «Размер»

Для того, чтобы классы «Двери пассажирские» и «Люки» так же содержали подкласс «Размер», выделите «Размер» в дереве иерархии. В окне Description в строке Superclasses нажмите . В появившемся окне выберете закладку «Class hierarchy» и при нажатой клавише Ctrl выделите классы «Двери грузовые», «Двери пассажирские» и «Люки» и нажмите ОК (Рисунок 3.12).

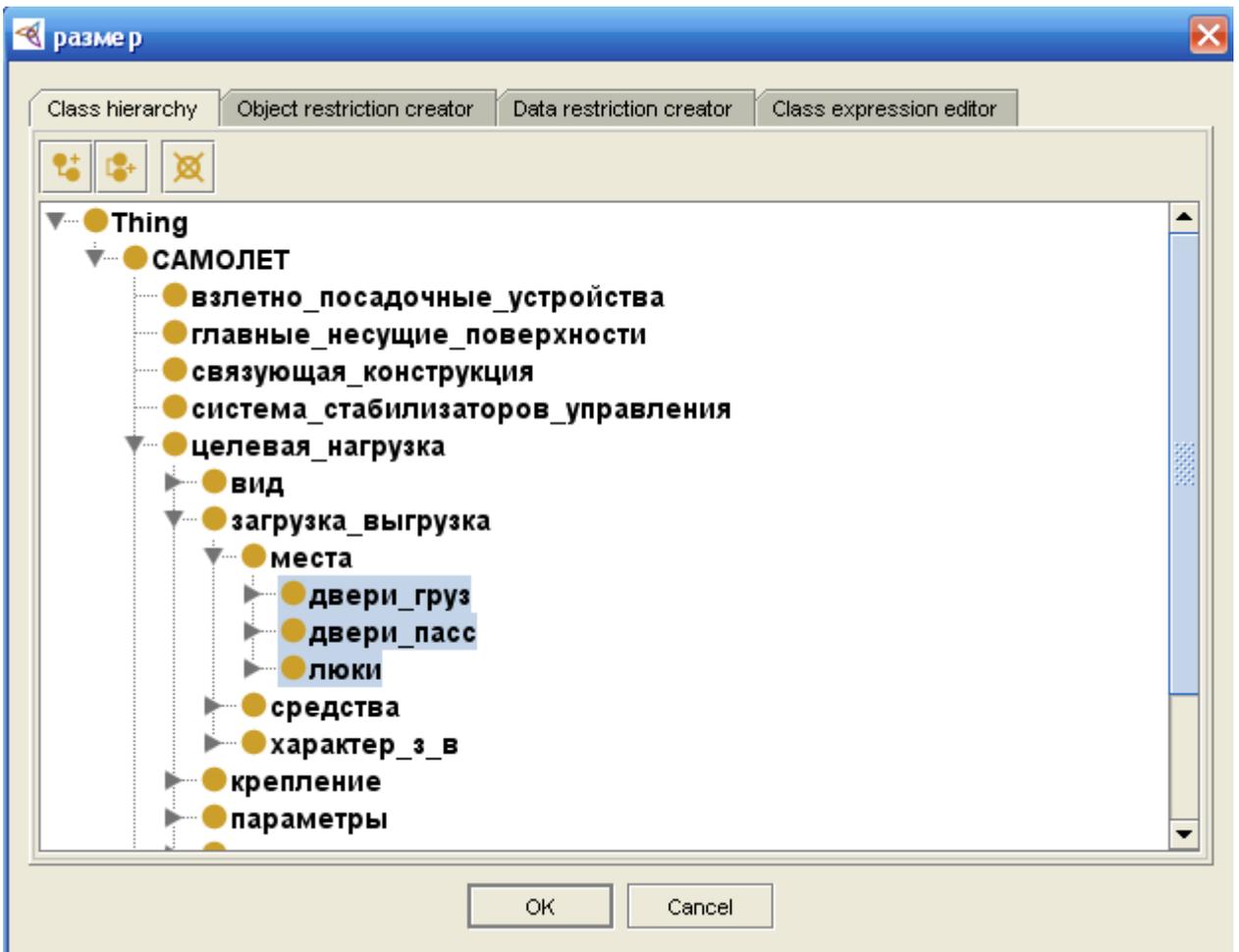


Рисунок 3.12 – Окно выбора суперклассов

После этого в строке Superclasses появятся классы, которые содержат подкласс «Размер» (Рисунок 3.13).

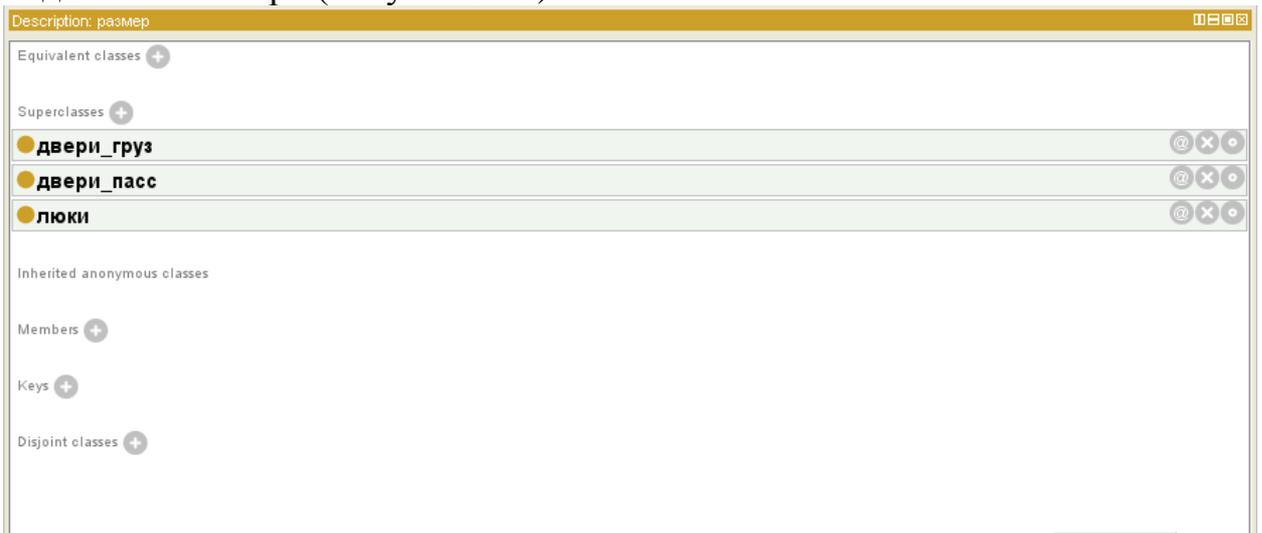


Рисунок 3.13 – Суперклассы подкласса «Размер»

Подклассы «Количество» и «Расположение» (Рисунок 3.14) так же должны содержаться в классах «Двери грузовые», «Двери пассажирские» и «Люки».

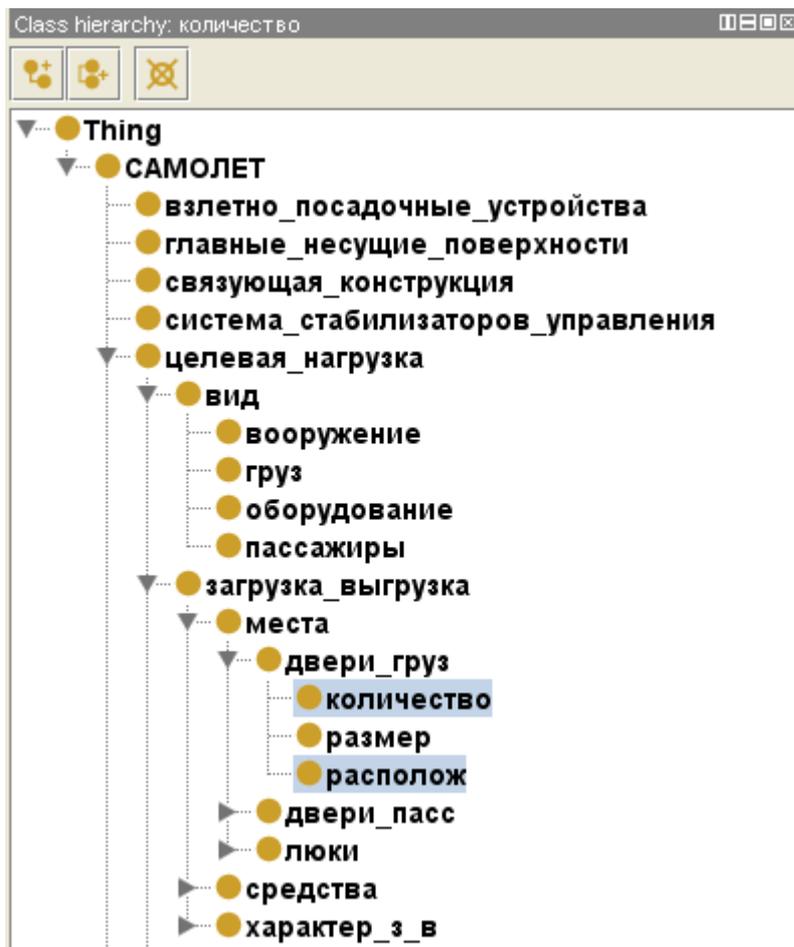


Рисунок 3.14 – Подклассы «Количество» и «Расположение»

Аналогичным образом включите подклассы «Количество» и «Расположение» в классы «Двери грузовые», «Двери пассажирские» и «Люки».

### 3.2.7. Создание свойств объекта и данных об объекте

Создание свойств и данных об объекте аналогично созданию классов.

В закладке «Object Properties» (Рисунок 3.15) с помощью кнопки  Вы можете добавлять свойства объектов.

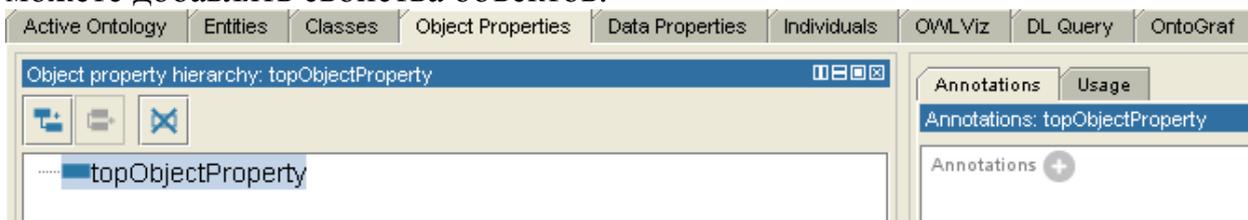


Рисунок 3.15 – Закладка «Object Properties»

В закладке «Data Properties» (Рисунок 3.16) с помощью кнопки  Вы можете добавлять данные об объекте.

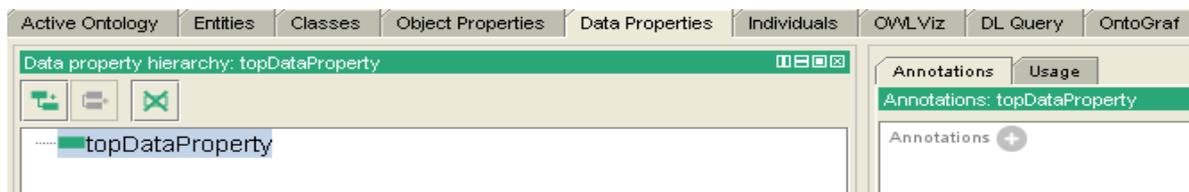


Рисунок 3.16 – Закладка «Data Properties»

Ранее был создан класс «Размер», необходимо уточнить какие именно размеры будут использованы при создании экземпляров онтологии. Для примера, в качестве размеров для таких объектов, как двери и люки, задайте параметры ширины и высоты. Для этого в закладке «Data Properties» с помощью кнопки  создайте свойства «ширина люка», «ширина пассажирской двери», «высота люка», и «высота пассажирской двери» (Рисунок 3.17).

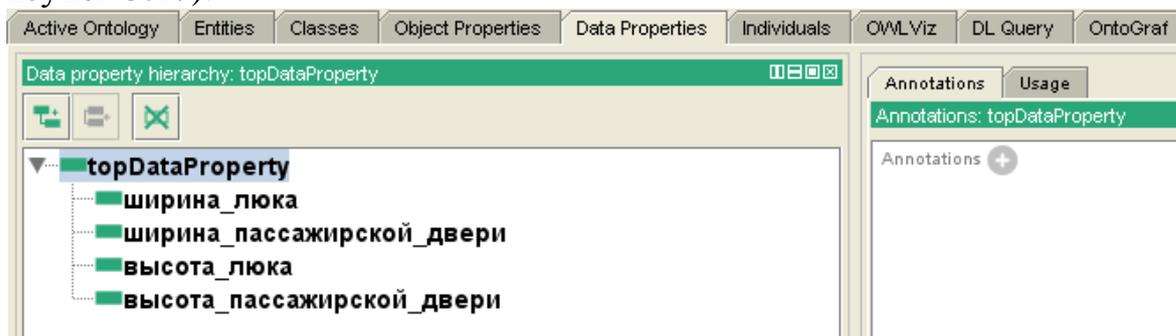


Рисунок 3.17 – Создание свойств «ширина» и «высота»

### 3.2.8. Создание экземпляров

Создание экземпляров в Protege 4.1 beta и в Protege 3.3 аналогично. Для создания экземпляра перейдите во вкладку «Individuals» (Рисунок 3.18).

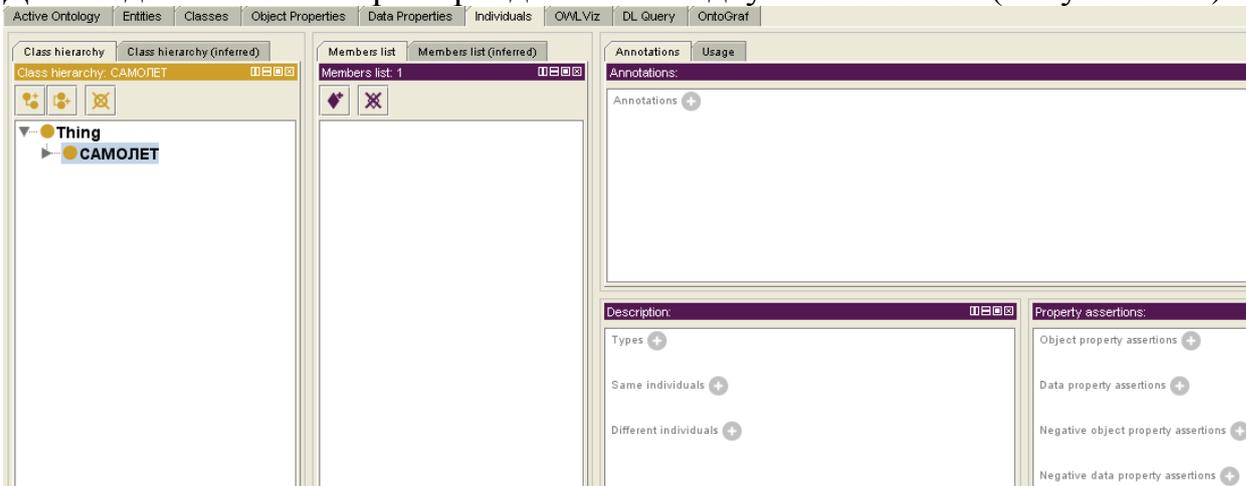


Рисунок 3.18 – Окно создания экземпляров

В дереве иерархии выберите класс «САМОЛЕТ» и нажмите кнопку  в окне создания экземпляров. В появившемся окне введите имя «Ил-76тд» и нажмите ОК (Рисунок 3.19).



Рисунок 3.19 – Окно ввода имени экземпляра

Онтология описывает всю предметную область. Создаваемые же экземпляры могут содержаться не во всех классах онтологии. Например, в онтологии содержатся классы «Вооружение», «Груз», «Оборудование», «Пассажиры», а в качестве экземпляра создается грузовой самолет. В этом случае, созданный экземпляр включать в класс «Пассажиры» не следует.

Созданный экземпляр Ил-76тд необходимо включить в выбранные классы. В качестве примера будут рассмотрены *места загрузки-выгрузки целевой нагрузки*, то есть класс «Места» и все его подклассы.

Для того, чтобы осуществить выбор экземпляра для класса, перейдите во вкладку Classes, выберите класс «Места» и в окне Description, в строке Members нажмите на кнопку . В появившемся окне выберите Ил-76тд (Рисунок 3.20).

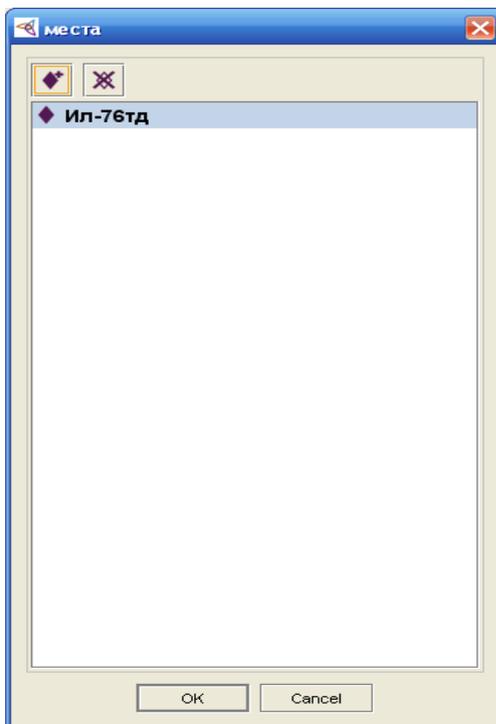


Рисунок 3.20 – Окно экземпляра для класса «Места»

После этого в окне Description, в строке Members появится наименование выбранного экземпляра (Рисунок 3.21).

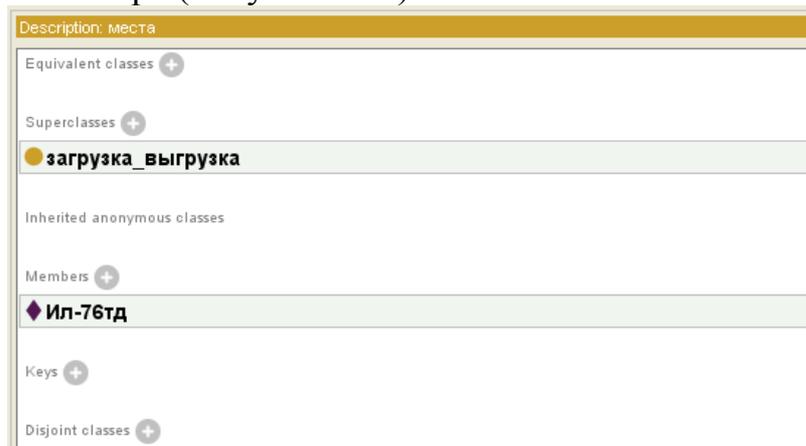


Рисунок 3.21 – Окно Description: места

### 3.2.9. Задание значений для экземпляров

Для того чтобы задать значения для созданного экземпляра выберите в дереве иерархии подкласс «Размер» класса «Двери пассажирские». В окне Description, в строке Superclasses нажмите кнопку . В появившемся окне перейдите во вкладку «Data restriction creator», в левом окне выберите «высота\_пассажирской\_двери», в правом – тип значения «integer». Из выпадающего списка находящегося внизу окна выберите «Exactly», введите значения высоты пассажирской двери самолета равное 1900 и нажмите ОК (Рисунок 3.22).

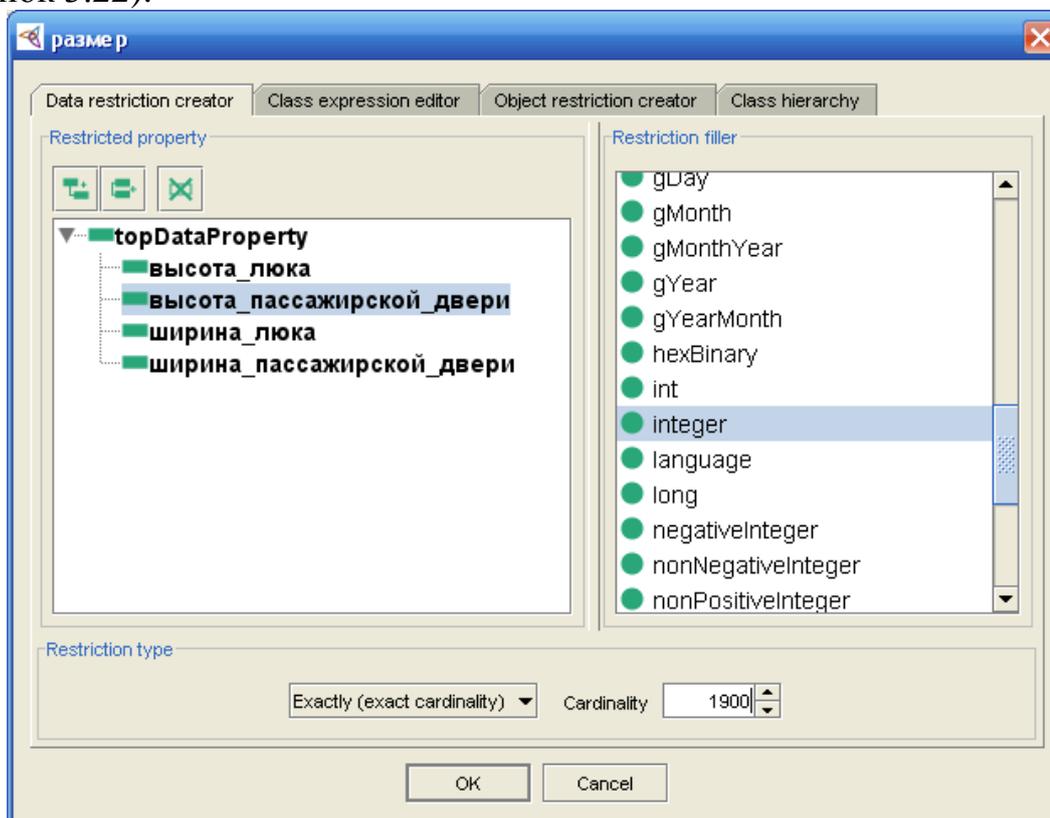


Рисунок 3.22 – Ввод значения высоты пассажирской двери самолета Ил-76тд

В этом окне Вы можете выбирать тип значения (integer, name, boolean и др.) и ограничения (Min, Max и др.).

Аналогичным образом введите значения ширины пассажирской двери самолета Ил-76тд равное 860 (Рисунок 3.23).

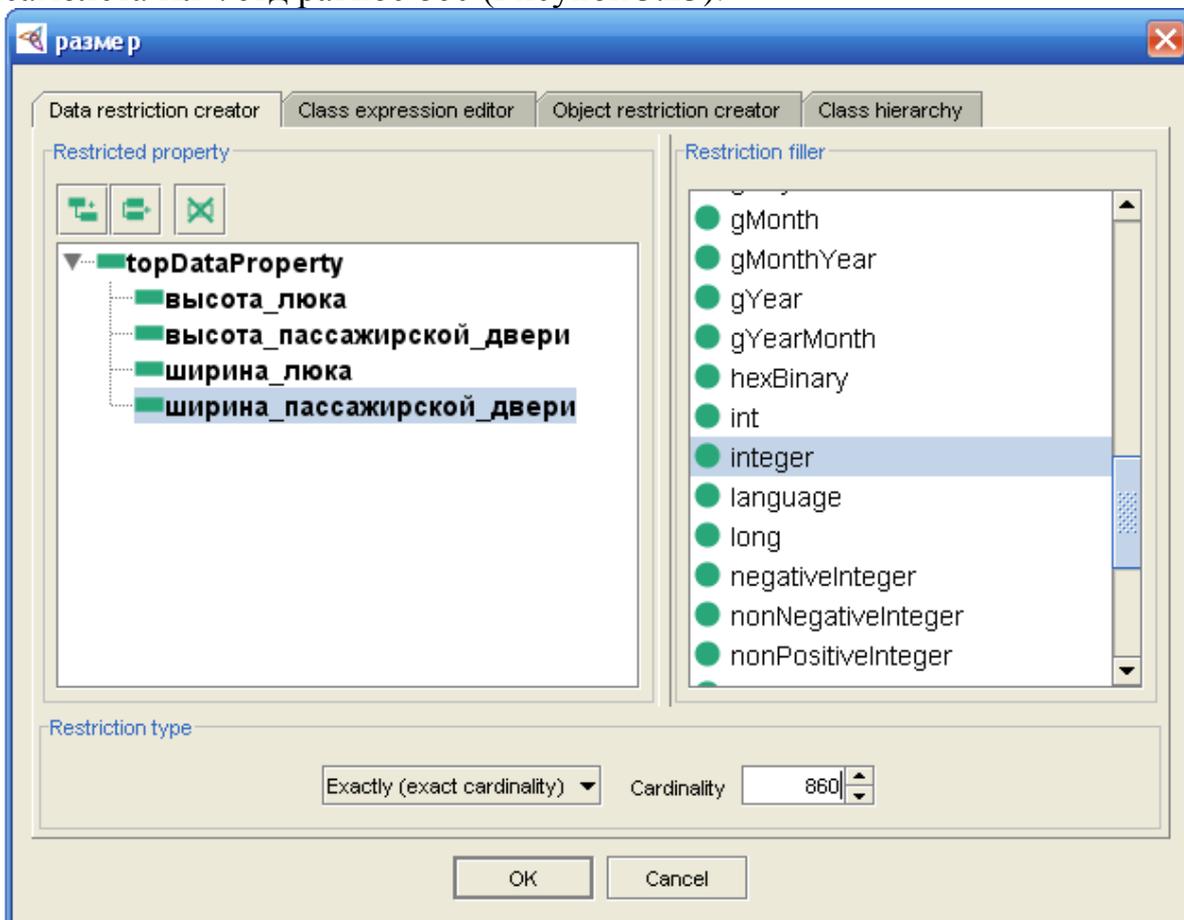


Рисунок 3.23 – Ввод значения ширины пассажирской двери самолета Ил-76тд

После этого окно Description: размер будет иметь вид, представленный на рисунке 3.24.

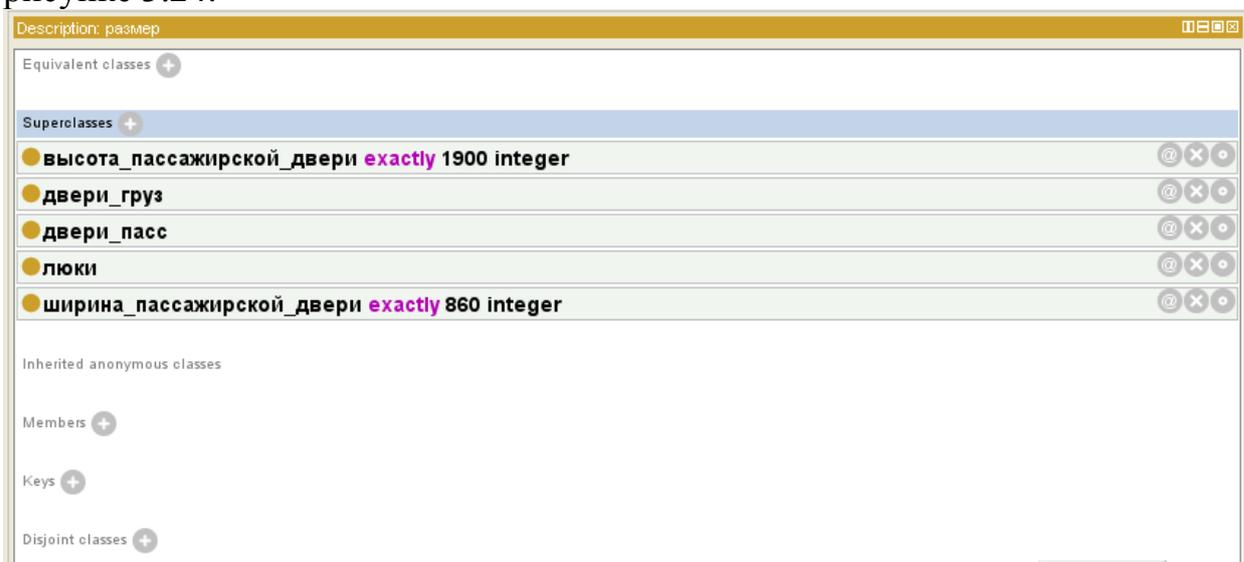


Рисунок 3.24 – Окно Description: размер

### 3.2.10. Создание семантического представления онтологии

Для создания семантического представления перейдите во вкладку OntoGraf и в дереве иерархии выберите «САМОЛЕТ». В окне справа появится кнопка «Самолет» (Рисунок 3.25).

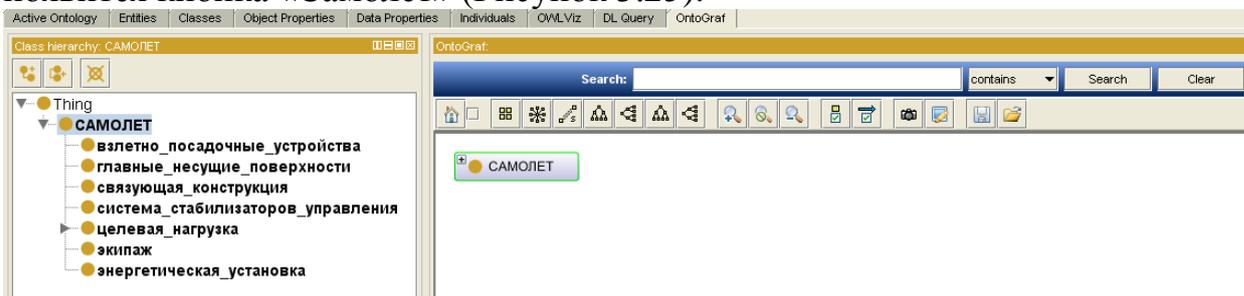


Рисунок 3.25 – Создание семантического представления

Знак «+» в левом верхнем углу кнопки «САМОЛЕТ» означает, что класс «Самолет» содержит подклассы или является подклассом. Кликните два раза на кнопку «САМОЛЕТ» для раскрытия структуры и получения фрагмента семантической сети (Рисунок 3.26).

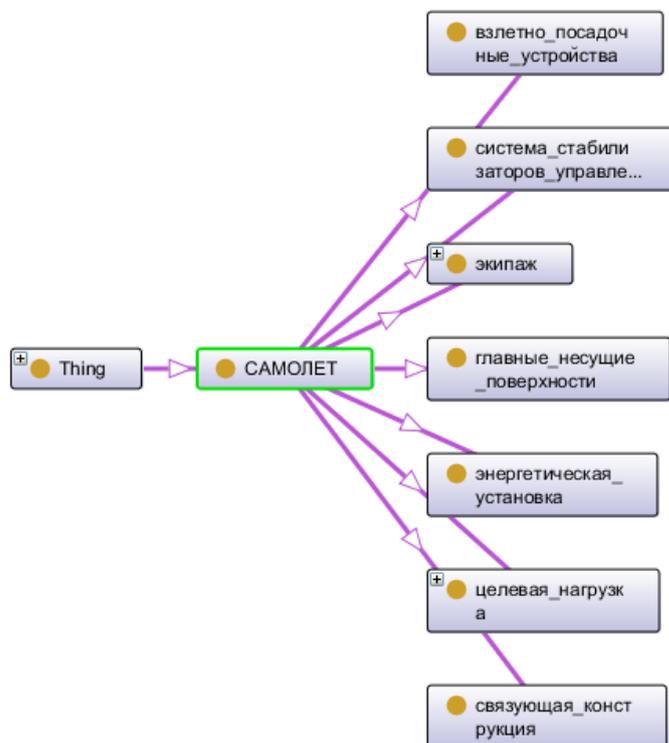


Рисунок 3.26 – Фрагмент семантической сети

Для того, чтобы раскрыть сеть, можно также воспользоваться иерархическим деревом, кликая на интересующий Вас объект.

Обратите внимание, что стрелки между классами и экземплярами имеют разный цвет (Рисунок 3.27).

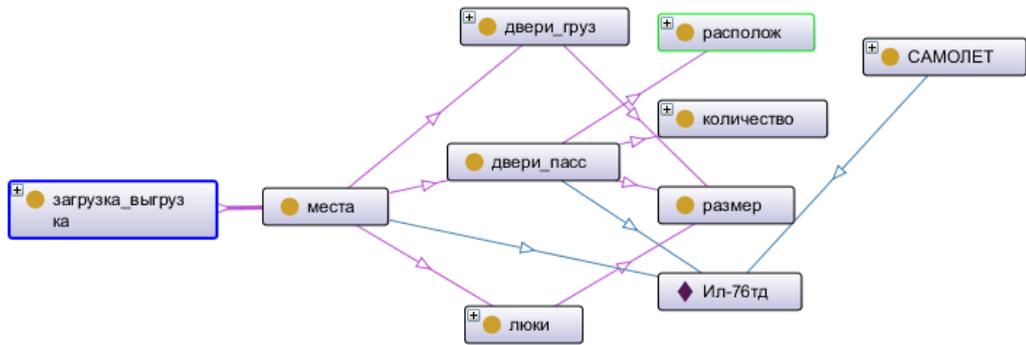


Рисунок 3.27 – Фрагмент семантической сети содержащей классы и экземпляры

### 3.2.11. Сохранение семантической сети

Для того чтобы сохранить семантическую сеть в формате JPEG, в окне OntoGraf нажмите кнопку , в появившемся окне выберете место сохранения, формат, введите название и нажмите ОК. Семантическая сеть представлена на рисунке 3.28.

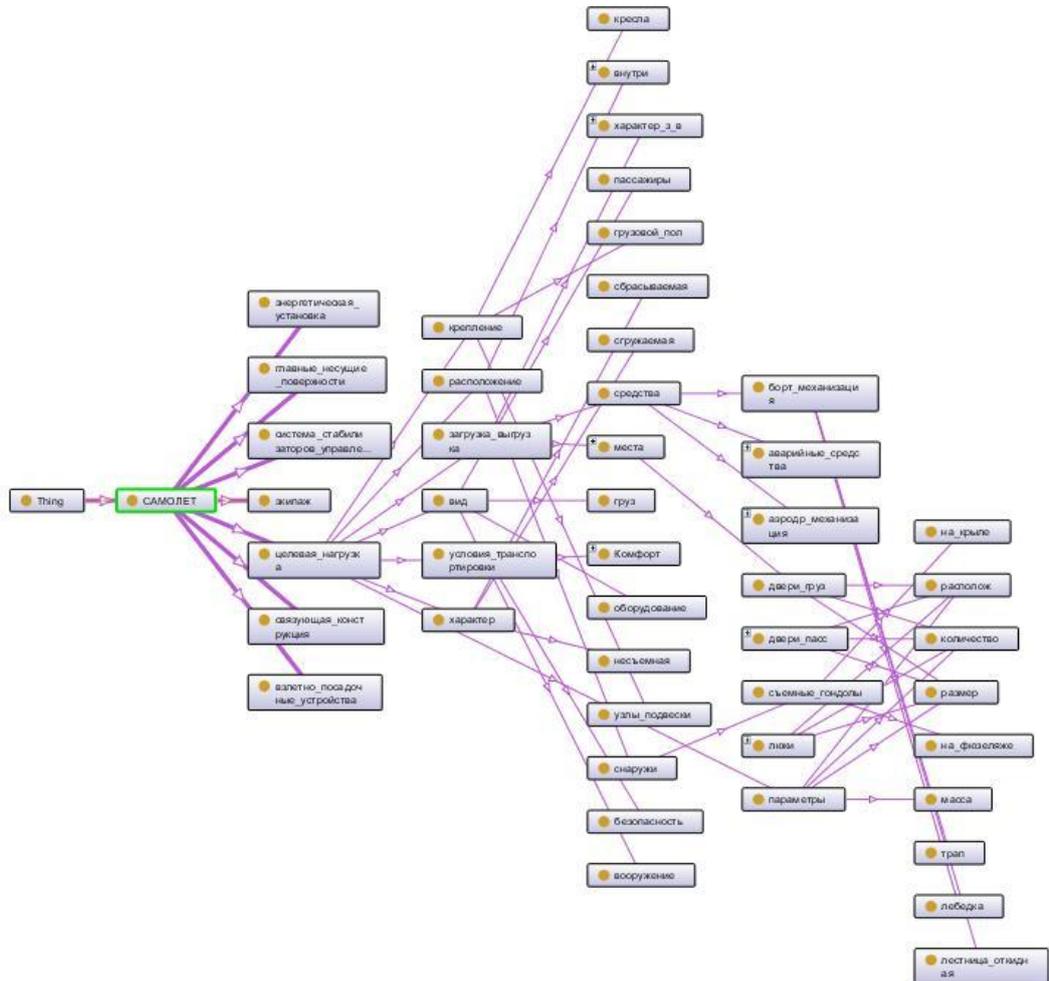
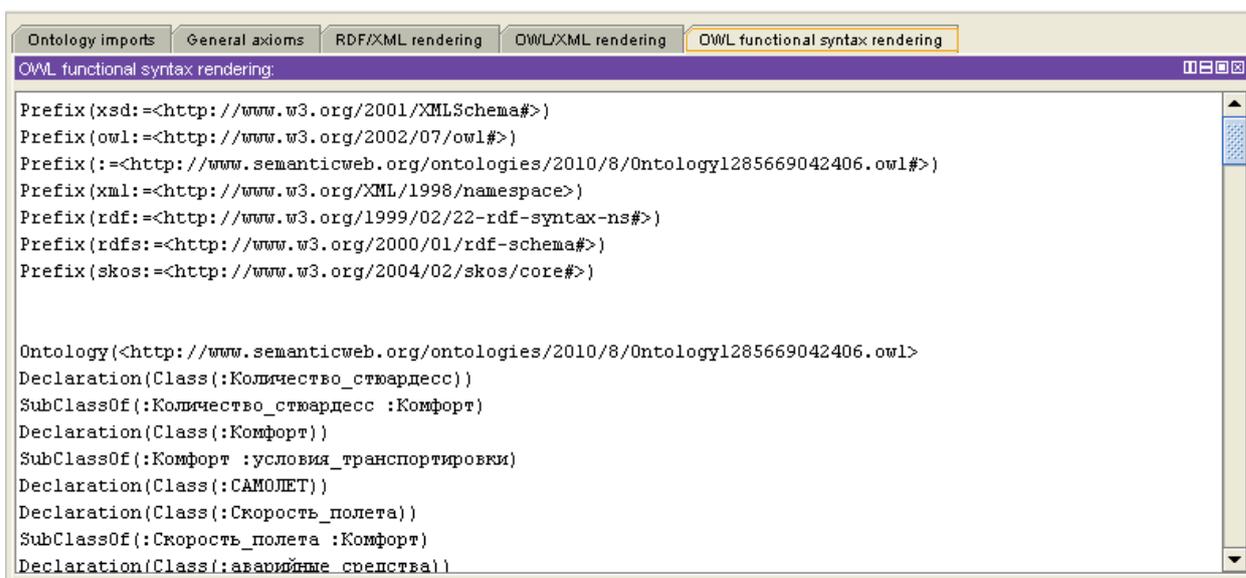


Рисунок 3.28 – Семантическая сеть

Также важным результатом работы в **Protégé** является возможность синтаксического представления в виде программы на OWL. Текст программы можно использовать и в других редакторах онтологий, имеющих возможность работы с OWL. Для получения синтаксического представления перейдите на закладку Active Ontology. В нижнем левом окне выберите вкладку «OWL functional syntax rendering» (Рисунок 3.29). Скопируйте и сохраните текст программы.



```
Ontology imports | General axioms | RDF/XML rendering | OWL/XML rendering | OWL functional syntax rendering
```

OWL functional syntax rendering:

```
Prefix (xsd: =<http://www.w3.org/2001/XMLSchema#>)
Prefix (owl: =<http://www.w3.org/2002/07/owl#>)
Prefix (: =<http://www.semanticweb.org/ontologies/2010/8/Ontology1285669042406.owl#>)
Prefix (xml: =<http://www.w3.org/XML/1998/namespace>)
Prefix (rdf: =<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix (rdfs: =<http://www.w3.org/2000/01/rdf-schema#>)
Prefix (skos: =<http://www.w3.org/2004/02/skos/core#>)

Ontology(<http://www.semanticweb.org/ontologies/2010/8/Ontology1285669042406.owl>
Declaration(Class(:Количество_стюардесс))
SubClassOf(:Количество_стюардесс :Комфорт)
Declaration(Class(:Комфорт))
SubClassOf(:Комфорт :условия_транспортировки)
Declaration(Class(:САМОЛЕТ))
Declaration(Class(:Скорость_полета))
SubClassOf(:Скорость_полета :Комфорт)
Declaration(Class(:аварийные_средства))
```

Рисунок 3.29 – Синтаксическое представление онтологии

### 3.3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ

1. Расширить созданную онтологию в соответствии с полученным вариантом задания (Таблица 3.1).
  - создать не менее трех уровней в дереве иерархии, начиная с класса по заданию,
  - создать не менее пяти подклассов для каждого из созданных уровней.
2. Наполнить онтологию значениями экземпляров (3-5 экз.).
3. Представить результат в виде семантической сети.
4. Включить в отчет синтаксическое представление созданной онтологии.

Таблица 3.1 – Варианты заданий

№ варианта	Элемент самолета
1	Взлетно-посадочные устройства
2	Главные несущие поверхности
3	Система стабилизаторов управления
4	Связующая конструкция
5	Энергетическая установка
6	Вооружение
7	Оборудование
8	Вид целевой нагрузки
9	Условия транспортировки
10	Люки
11	Средства загрузки-выгрузки
12	Крепление целевой нагрузки
13	Расположение целевой нагрузки
14	Расположение несущих поверхностей
15	Вид силовой установки

#### Литература

1. Боргест Н.М. Онтология проектирования: теоретические основы. Учеб. пособие. Самара: СГАУ, 2010 – 88 с.
2. Боргест Н.М. Антология онтологии: подборка научных статей. Самара: СГАУ, 2010 – 88 с.
3. Боргест Н.М., Симонова Е.В. Основы построения мультиагентных систем, использующих онтологию. Учеб. пособие. Самара: СГАУ, 2009 – 80 с.
4. Боргест Н.М. Автоматизация предварительного проектирования самолета. Учеб. пособие. Самара: Самар. авиац. ин-т, 1992, 92с.
5. Проектирование самолетов: учебник для вузов/С.М. Егер, В.Ф. Мишин, Н.К. Лисейцев и др. Под ред. С.М. Егера. - М.: Логос, 2005. - 648 с.