

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЕВА» (СГАУ)

**ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ ПРИ ВЫБОРЕ САМОЛЕТА
ПОД ЗАДАННОЕ ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Самара
2008

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С. П. КОРОЛЕВА» (СГАУ)

**ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ ПРИ ВЫБОРЕ САМОЛЕТА
ПОД ЗАДАННОЕ ТЕХНИЧЕСКОЕ ЗАДАНИЕ**
Методические указания к лабораторной работе № 1

Самара
Издательство СГАУ
2008

Составители: Н.М. Боргест, Е.В. Симонова

УДК 681.3

Использование онтологии при выборе самолета под заданное техническое задание: метод. указания / Сост. *Н.М. Боргест, Е.В. Симонова.* – Самара: Изд-во Самар. гос. аэрокосм. ун-та. 2008. - 61с.

Методические указания описывают основные этапы решения задачи подбора самолетов-прототипов, чьи характеристики отвечают ТТТ к проектируемому самолету, с использованием онтологии: освоение приемов проектирования онтологии предметной области «Выбор самолета под заданное техническое задание», освоение приемов конструирования и моделирования онтологической сцены.

Методические указания предназначены для студентов, обучающихся по специальности 220305 – «Автоматизированное управление жизненным циклом продукции», при изучении курса «Онтология производственной сферы».

Разработаны на кафедре конструкции и проектирования летательных аппаратов совместно с кафедрой информационных систем и технологий. Методические указания базируются на закупленном в рамках ИОП программном обеспечении, предоставленном компанией Magenta Development.

Печатаются по решению редакционно-издательского совета Самарского государственного аэрокосмического университета

Рецензент: *Коварцев А.Н.*

1. ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ.....	6
2. ВЫБОР ТИПА САМОЛЕТА ПО КРИТЕРИЮ ДАЛЬНОСТИ ПОЛЕТА.....	7
2.1. ПОСТАНОВКА ЗАДАЧИ	7
2.2 ПРОЕКТИРОВАНИЕ ДЕСКРИПТИВНОЙ ОНТОЛОГИИ.....	8
2.2.1 <i>Создание онтологии</i>	8
2.2.2 <i>Создание и удаление концепта</i>	9
2.2.2.1 Концепт «объект»	9
2.2.2.2 Концепт «атрибут»	15
2.2.2.3 Ограничения на значения атрибутов	17
2.2.2.4 Создание связей между концептами	18
2.2.2.5 Представление онтологии в виде семантической сети	20
2.2.3 <i>Концепт «скрипт»</i>	21
2.2.3.1 Определение скриптов для вычисления границ поиска самолета-прототипа в базе данных	22
2.3 ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ МИРА ЗАКАЗОВ И РЕСУРСОВ.....	25
2.3.1 <i>Создание онтологии мира заказов и ресурсов</i>	25
2.3.2 <i>Создание концепта «агент заказа»</i>	26
2.3.3 <i>Создание концепта «агент ресурса»</i>	28
2.3.4 <i>Виртуальные отношения: отношение матчинга</i>	30
2.3.5 <i>Условия матчинга</i>	32
2.3.5.1 Создание условий матчинга, ограничивающих поиск самолетов-прототипов в базе данных	34
2.3.6 <i>Условия принятия решения (Decision Making Machine conditions)</i>	36
2.3.6.1 Создание условия принятия решения - максимизация дальности полета самолета-прототипа	37
2.3.7 <i>Параметры представления экземпляра объекта в сцене</i>	38
2.3.8 <i>Сохранение онтологий предметной области “Тактико-технические требования к проектируемому самолету”</i>	40
2.4 СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ СЦЕНЫ	41
2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА	44
2.5.1 <i>Запуск сцены на моделирование</i>	44
2.5.2 <i>Сохранение сцены виртуального мира</i>	47
2.5.3. <i>Загрузка ранее созданной сцены виртуального мира</i>	48
2.5.4 <i>Изучение возможностей системного лога</i>	48
2.5.5 <i>Изменение значений атрибутов агентов в сцене</i>	50
2.5.6 <i>Добавление агентов в сцену</i>	51
3. ВЫБОР ТИПА САМОЛЕТА ПО КРИТЕРИЮ ВЗЛЕТНОЙ МАССЫ	52
3.1. ПОСТАНОВКА ЗАДАЧИ	52
3.2 ПРОЕКТИРОВАНИЕ ДЕСКРИПТИВНОЙ ОНТОЛОГИИ.....	53
3.2.1 <i>Создание онтологии</i>	53
3.2.2 <i>Редактирование дерева концептов онтологии</i>	54

3.3 ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ МИРА ЗАКАЗОВ И РЕСУРСОВ	55
3.3.1 Редактирование условий матчнга	55
3.3.2 Создание условия принятия решения - минимизация взлетной массы самолета-прототипа	56
3.3.3 Сохранение онтологий предметной области “ТТТ к проектируемому самолету”	57
3.4 СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ СЦЕНЫ	57
3.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА	58
4 КОНТРОЛЬНЫЕ ВОПРОСЫ	60
5 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....	61

1. ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

Разработке тактико-технических требований (ТТТ) к проектируемому самолету предшествует весьма важный и трудоемкий комплекс предварительных исследований, называемый анализом проектной ситуации, который в значительной мере основывается на изучении статистического материала. Статистика позволяет проследить историю развития самолетов данного типа, установить численные значения основных параметров и летно-технических показателей этих самолетов и выявить тенденции их дальнейшего развития.

Статистические таблицы, составляемые при разработке нового самолета, содержат сведения об основных характеристиках и параметрах самолетов-прототипов, идентичных по назначению и условиям применения проектируемому самолету, и имеющих примерно одинаковые с ним целевую нагрузку и дальность полета.

При отборе самолетов для включения в статистику следует иметь в виду, что летные характеристики и относительные параметры самолетов не очень сильно зависят от их абсолютных размеров и масс. Это позволяет включать в статистику прототипы, которые по массе целевой нагрузки, дальности полета могут существенно (до 30-40%) отличаться от показателей проектируемого самолета. Это расширяет возможности для сбора статистического материала.

Источниками для сбора статистики могут служить как отечественные, так и зарубежные справочники, энциклопедии по авиации, отчеты НИИ, журналы и другая литература. Все большее развитие получают компьютерные базы данных по авиационной технике.

ТТТ к проектируемому самолету определяют основные цели и задачи его создания, условия его применения, задают потребные значения основных параметров и характеристик самолета, намечают условия его производства и эксплуатации.

Все требования к проектируемому самолету подразделяются на несколько групп. Они представляются двояким образом. Во-первых, составляется перечень основных, наиболее важных для данного самолета требований, носящих качественный характер, без указания каких-либо численных их значений. И, во-вторых, задаются количественные требования с указанием численных значений или ограничений для летно-технических параметров и характеристик – летно-технические требования.

Основные характеристики самолетов-прототипов, а также требования к проектируемому самолету могут быть описаны в виде *онтологии*. При этом проектируемый самолет следует рассматривать как *проект* или *заказ*, а самолеты-прототипы, характеристики которых указаны в базе данных, – как *ресурсы*. 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 выполняется подбор самолетов-прототипов, чьи характеристики отвечают ТТТ к проектируемому самолету.

Цель лабораторной работы № 1 – разработка мультиагентных приложений, реализующих простые возможности одностороннего матчинга в процессе принятия решений.

В процессе выполнения лабораторной работы № 1 решаются следующие задачи:

- Освоение инструментов, предоставляемых конструктором онтологий и исполняющей системой.
- Освоение приемов проектирования дескриптивной онтологии и онтологии мира заказов/ресурсов.
- Освоение приемов конструирования и моделирования онтологической сцены.
- Изучение матчинга между агентами заказов/ресурсов и процесса принятия решения агентом заказа по резервированию агента ресурса.

В начале каждого раздела излагаются общие теоретические положения. Действия, которые непосредственно предлагается выполнить пользователю, выделяются знаками, показанными ниже. Будьте, пожалуйста, внимательны!



2. ВЫБОР ТИПА САМОЛЕТА ПО КРИТЕРИЮ ДАЛЬНОСТИ ПОЛЕТА

2.1. Постановка задачи

На основании заданных тактико-технических требований для проектируемого самолета необходимо подобрать в базе данных самолет, отвечающий заданным требованиям. Назначение самолета – пассажирский, либо грузовой.

ТТТ для пассажирского самолета:

- дальность полета, км;
- число пассажиров, чел.

ТТТ для грузового самолета:

- дальность полета, км;
- масса целевой нагрузки, кг;

Задан допустимый процент отклонений тактико-технических требований прототипа от ТТТ проектируемого самолета.


Необходимо выбрать в базе данных прототипы, отвечающие ТТТ по дальности полета и количеству пассажиров. Среди этих прототипов

необходимо выбрать самолет, обеспечивающий максимальную дальность полета.

Ниже рассмотрен пример выбора пассажирского самолета. Выбор грузового самолета производится аналогично.

2.2 Проектирование дескриптивной онтологии

2.2.1 Создание онтологии

После загрузки конструктора онтологий для создания новой онтологии необходимо воспользоваться пунктом меню *File* → *New*, сочетанием клавиш $\langle Ctrl \rangle + \langle N \rangle$ или выбрать кнопку  на рабочей панели. На экране появляется библиотека онтологий (рис. 1).

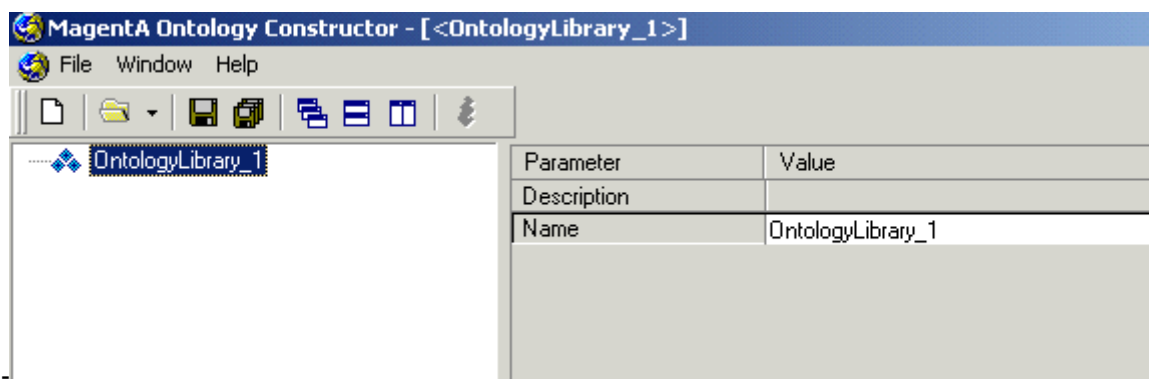


Рис. 1. Создание библиотеки онтологий

Создание новой онтологии в библиотеке осуществляется посредством выделения библиотеки в левой части экрана и нажатия правой кнопки мыши. В открывшемся контекстном меню следует выбрать пункт *New Item (Новый элемент)*, после чего появится возможность создания онтологий двух типов: *Descriptive ontology* (Описательная онтология) и *Virtual World ontology* (Онтология мира заказов и ресурсов) (рис. 2). *Descriptive ontology* предназначена для описания предметной области и оперирует такими понятиями, как объект, атрибут, скрипт и т.п. *Virtual World ontology* описывает мир в терминах заказов, ресурсов и отношений матчинга между ними.

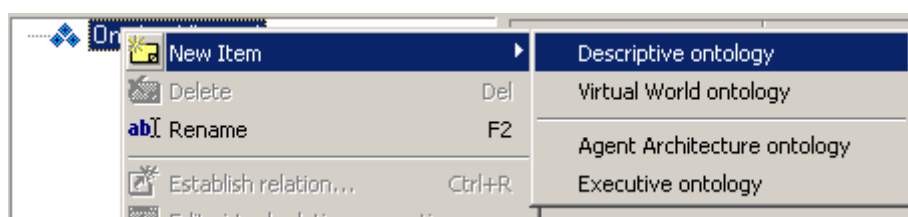


Рис. 2. Создание онтологии предметной области

- ◆
- ◆ 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_TTT*.
- ◆ Создайте дескриптивную онтологию предметной области «Тактико-технические требования к самолету» (*New Item → Descriptive ontology*). По умолчанию дескриптивная онтология имеет имя *Ontology_1*. Переименуйте ее, вводя в поле *Name* имя *Ontology_TTT_Distance*.
- ◆

2.2.2 Создание и удаление концепта

После создания онтологии нажатием на кнопку <+> открывается 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*. При этом вся иерархия классов поддерживает наследование свойств (создав, например, атрибут строкового типа, можно создать его потомков, и все они тоже будут строковыми). Дерево категорий концептов дескриптивной онтологии показано на рис. 3.

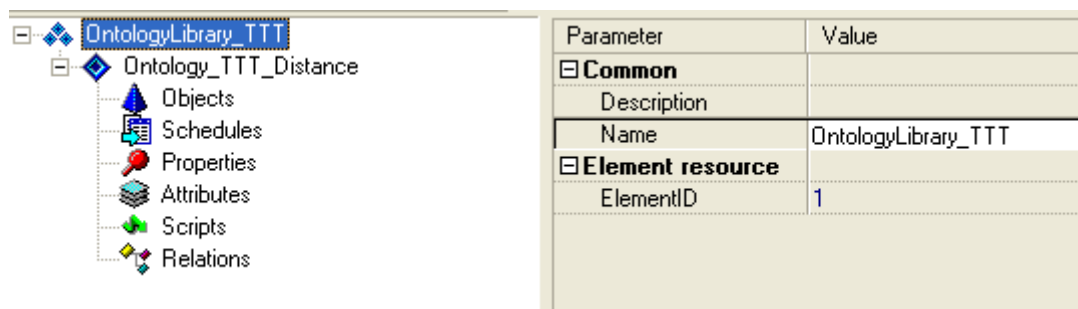


Рис. 3. Дерево категорий концептов дескриптивной онтологии

Созданный концепт можно удалить, выделив его и нажав клавишу , либо выбрав в контекстном меню пункт *Delete*.

2.2.2.1 Концепт «объект»

Концепт «объект» - это сущность, которая присутствует в мире, описанном в онтологии. После создания дескриптивной онтологии ПО «Тактико-технические требования к самолету» необходимо создать два

концепта «объект»:

- *Project_Airplane* (проектируемый самолет) с атрибутами
 - *Project_Distance* (требуемая дальность полета);
 - *Percent of the Deviation_Distance* (допустимый процент отклонений от требуемой дальности полета при выборе прототипа);
 - *Project_Min Distance* (минимальная требуемая дальность полета);
 - *Project_Max Distance* (максимальная требуемая дальность полета);
 - *Project_Number of Passengers* (требуемое количество пассажиров);
 - *Percent of the Deviation_Number of Passengers* (допустимый процент отклонений от требуемого количества пассажиров при выборе прототипа);
 - *Project_Min Number of Passengers* (минимальное требуемое количество пассажиров);
 - *Project_Max Number of Passengers* (максимальное требуемое количество пассажиров).
- *BD_Airplane* (самолет-прототип в базе данных) с атрибутами
 - *BD_Name of the Plane* (тип самолета);
 - *BD_Distance* (дальность полета);
 - *BD_Number of Passengers* (количество пассажиров).

Необходимо задать также координаты X и Y для представления экземпляра концепта «объект» (агента) в сцене.

Для создания концепта «объект» необходимо выделить в дереве категорий категорию *Objects*, в контекстном меню выбрать *New Item -> Object* (Рис. 4).

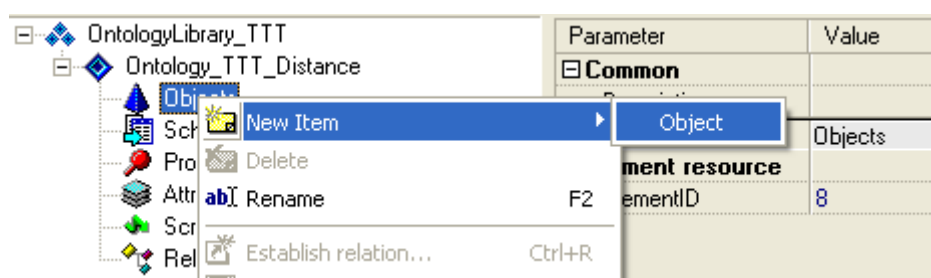


Рис. 4. Создание концепта объект (*Object*)

Концепт категории «объект» имеет следующие свойства (параметры) (рис. 5):

- *Common* – словесное описание (*Description*), тип (*Element Type*), имя концепта (*Name*).
- *Element resource* – характеристики пиктограммы для представления агента концепта «объект» в сцене.

- *ElementID* – зарезервированный атрибут, изменять его не следует;
- *Extra large icon*, *Large icon*, *Small icon* – три вида пиктограмм для представления концепта в сцене.
- *Interface options*
 - *Add to palette* – запрет/разрешение на добавление пиктограммы концепта «объект» в палитру при отображении в сцене,
 - *Interface behaviour* - группа свойств, определяющих поведение агента концепта «объект» в сцене.

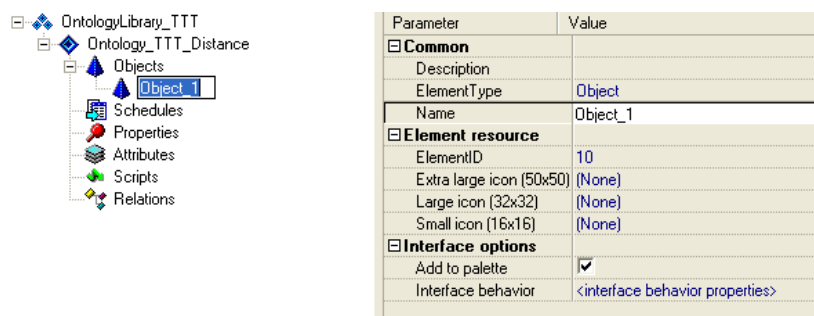


Рис. 5. Свойства концепта «объект»

Каждому концепту (в том числе, и объекту) можно назначить пиктограмму, с помощью которой данный концепт будет отображаться при работе со сценой в процессе моделирования. В редакторе свойств концепта в параметре *Element resource* -> *Small Icon* (либо *Large Icon* или *Extra large Icon*) следует вызвать диалог редактирования изображения (рис. 6). Для каждого из концептов, отображаемых в сцене, следует задать пиктограммы всех трех типов, т.к. во время создания онтологии размер сцены может быть еще неизвестен.



Рис. 6. Вызов диалога редактирования изображения концепта

Пиктограмму можно загрузить двумя способами:

- из хранилища иконок (*Picture Editor* -> *Load* -> *Load from repository*, в открывшемся окне *Select Icon* выбрать пиктограмму из списка -> *OK* -> подтвердить выбор в окне *Picture Editor* -> *OK*) (рис. 7-10).

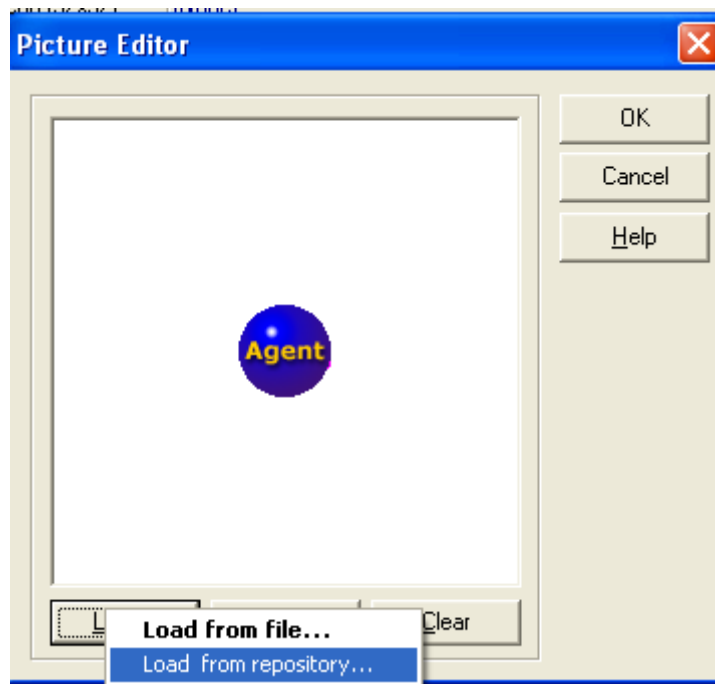


Рис. 7. Диалог выбора источника загрузки пиктограмм

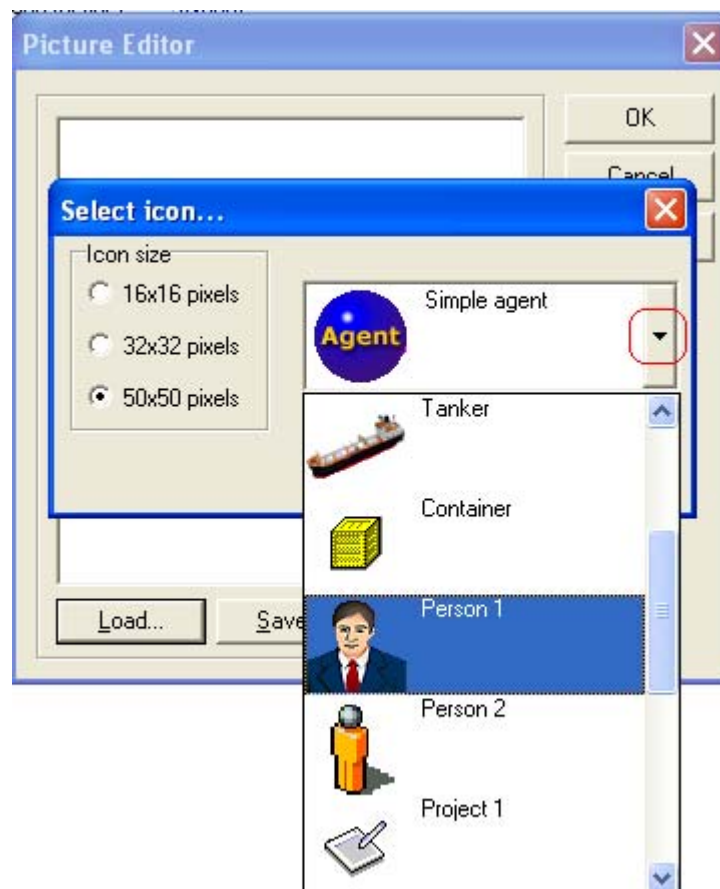


Рис. 8. Список пиктограмм в репозитории

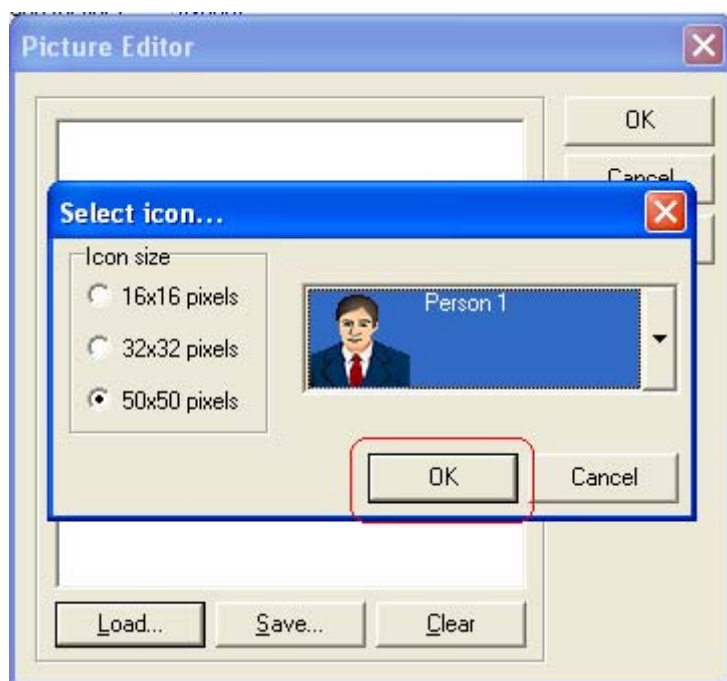


Рис. 9. Выбор пиктограммы из списка в репозитории



Рис. 10. Подтверждение загрузки пиктограммы из репозитория

- из файла, который пользователь предварительно подготовил и разместил в папке Pictures (*Picture Editor* -> *Load* -> *Load from file* , в открывшемся окне *Load Picture* перейти к папке, в которой находятся изображения допустимых типов (*jpg, jpeg, bmp, ico, emf, wmf*), затем в списке файлов, содержащих пиктограммы, выбрать нужный -> *Открыть* -> подтвердить выбор в окне *Picture Editor* -> *OK*) (рис. 11-12);

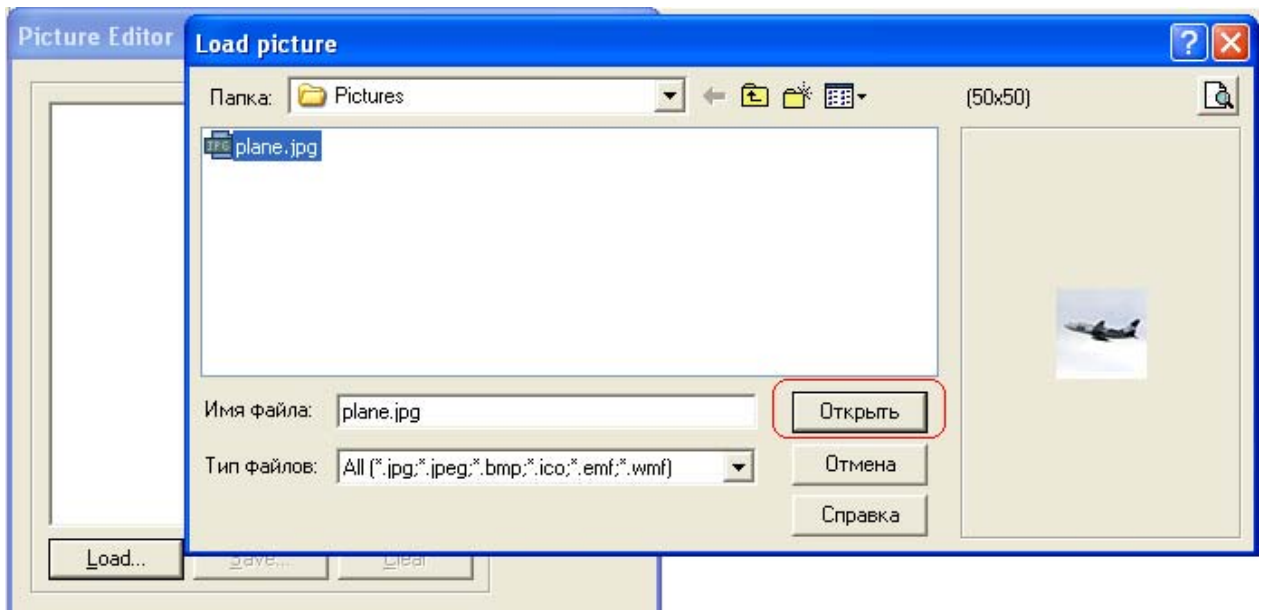


Рис. 11. Список файлов, содержащих пиктограммы

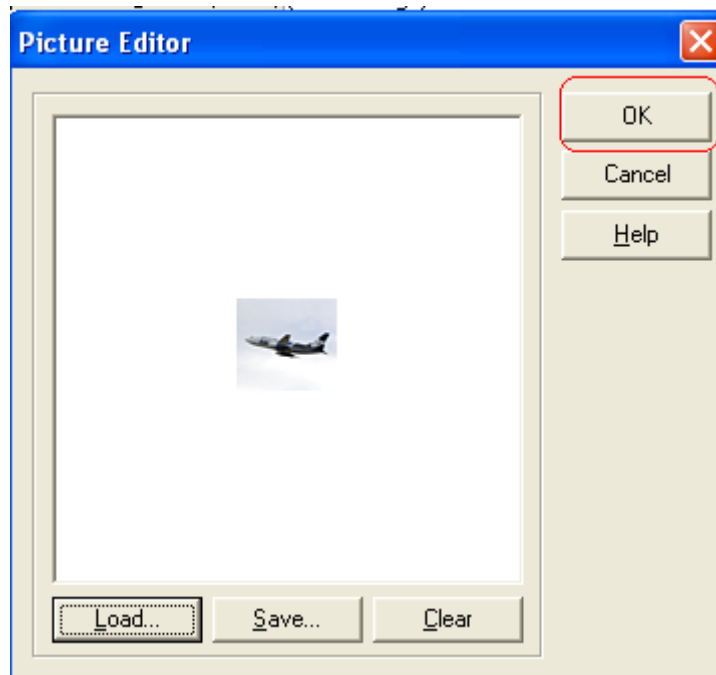


Рис. 12. Подтверждение загрузки пиктограммы из файла

- ◆
- ◆ Создайте объект *Project_Plane* (*Objects* -> *New Item* -> *Object*), переименуйте его в *Project_Plane*, изменяя значение в поле *Name*, назначьте этому объекту пиктограммы, соответствующие трем видам отображения (как показано на рис. 7-10 или на рис. 11-12).
- ◆ Создайте объект *BD_Plane* (аналогично созданию *Project_Plane*), переименуйте его в *BD_Plane*, назначьте этому объекту пиктограммы,

соответствующие трем видам отображения.



Каждый концепт «объект» может иметь определенный список атрибутов.

2.2.2.2 Концепт «атрибут»

Концепт «атрибут» – это величина, характеризующая объект (количественное выражение признака). Конструктор онтологий позволяет создавать атрибуты следующих типов:

- *Integer attribute* – целое число,
- *Float attribute* – вещественное число,
- *Currency attribute* – денежный атрибут,
- *String attribute* – строковый атрибут,
- *Boolean attribute* – булево значение,
- *Date attribute* – дата,
- *Date Time attribute* – дата и время,
- *Time attribute* – время,
- *Period attribute* – период,
- *String list attribute* – список строк,
- *Variant list attribute* – список величин типа variant,
- *Subagent attribute* – ссылка на зависимого агента (субагента),
- *Agent reference attribute* – уникальный идентификатор агента.

Для создания концепта «атрибут» необходимо выделить в дереве категорий категорию *Attributes*, в контекстном меню выбрать *New Item* -> тип атрибута (Рис. 13).



Рис. 13. Создание целочисленного атрибута

Концепт категории «атрибут» имеет следующие свойства (параметры) (рис. 14):

- *Common* – словесное описание (*Description*), тип (*Element Type*), имя концепта (*Name*).
- *Element resource* – характеристики пиктограммы для представления агента концепта «атрибут» в сцене.

- *ElementID* – зарезервированный атрибут, изменять его не следует;
- *Extra large icon, Large icon, Small icon* – три вида пиктограмм для представления концепта в сцене.
- *Other*
 - *Constraints* - ограничения на диапазон изменения значений атрибутов,
 - *Type name* – тип атрибута,
 - *Value* - значение атрибута. Для большинства атрибутов можно задать значение по умолчанию, которое будет устанавливаться при создании атрибутов в сцене. Для такой инициализации необходимо у данного атрибута отредактировать в редакторе свойств параметр *Value* (Значение). Значения по умолчанию наследуются у атрибутов-потомков, причем в потомке можно изменить унаследованное значение.

Parameter	Value
Common	
Description	
ElementType	Attribute
Name	Experience
User read only	<input type="checkbox"/>
Element resource	
Extra large icon (50x50)	(None)
Large icon (32x32)	(None)
Small icon (16x16)	(None)
Other	
Constraints	(TOntologyAttributeConstraints)
TypeName	Integer
Value	0

Рис. 14. Свойства концепта «атрибут»

- ◆
- ◆ Создайте целый атрибут *Project_Distance* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *Project_Distance*.
- ◆ Создайте вещественный атрибут *Percent of the Deviation_Distance* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *Percent of the Deviation_Distance*.
- ◆ Создайте целый атрибут *Project_Min Distance* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *Project_Min Distance*.
- ◆ Создайте целый атрибут *Project_Max Distance* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *Project_Max Distance*.
- ◆ Создайте целый атрибут *Project_Number of Passengers* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *Project_Number of Passengers*.
- ◆ Создайте вещественный атрибут *Project_Percent of the Deviation_Number of Passengers* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *Project_Percent of the Deviation_Number of Passengers*.

Passengers.

- ◆ Создайте целый атрибут *Project_Min Number of Passengers* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *Project_Min Number of Passengers*.
- ◆ Создайте целый атрибут *Project_Max Number of Passengers* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *Project_Max Number of Passengers*.
- ◆ Создайте строковый атрибут *BD_Name of thePlane* (*Attributes -> New Item -> String Attribute*), переименуйте созданный атрибут в *BD_Name of Plane*.
- ◆ Создайте целый атрибут *BD_Distance* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *BD_Distance*.
- ◆ Создайте целый атрибут *BD_Number of Passengers* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *BD_Number of Passengers*.
- ◆ Создайте целый атрибут *X* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *X*.
- ◆ Создайте целый атрибут *Y* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *Y*.
- ◆

2.2.2.3 Ограничения на значения атрибутов

Существует также возможность задать наследуемые ограничения на диапазон изменения значений атрибутов (не для всех типов атрибутов). Для определения ограничений следует выбрать необходимый атрибут и изменить его параметр *Constraints* (Ограничения) (рис. 15).

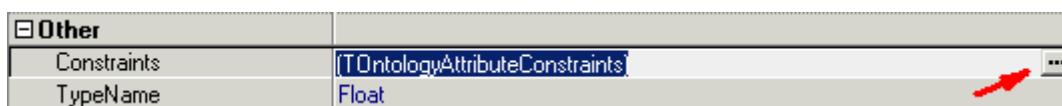


Рис. 15. Задание ограничений на значение атрибута

Далее, в открывшемся диалоговом окне можно задать ряд чисел, указывая допустимые пределы изменения значений (рис. 16).

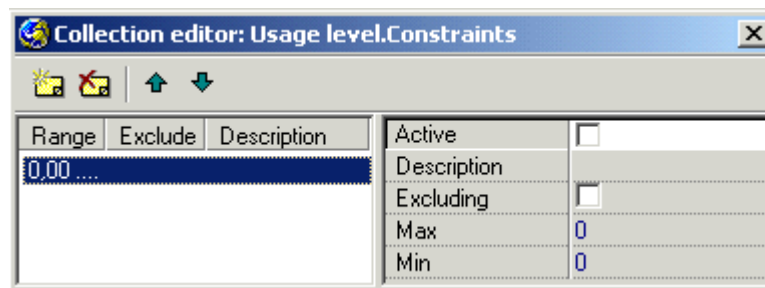



Рис. 16. Диалог задания ограничений на значение атрибута

С помощью кнопки  создается единичное интервальное ограничение, т.е., ограничение вида $x \in [a, b]$ либо $x \notin [a, b]$ (в зависимости от параметра *Excluding*). Более сложные ограничения можно строить суперпозицией единичных интервальных ограничений.

При вводе нового ограничения следует сначала ввести все параметры ограничения и только потом выставить флаг активности *Active*.

2.2.2.4 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 «Тактико-технические требования к самолету», т.е. указать, что объект *Project_Plane* имеет атрибуты *Project_Distance*, *Percent of the Deviation_Distance*, *Project_Min Distance*, *Project_Max Distance*, *Project_Number of Passengers*, *Percent of the Deviation_Number of Passengers*, *Project_Min Number of Passengers*, *Project_Max Number of Passengers*, *X*, *Y*, а объект *BD_Plane* – атрибуты *BD_Name of thePlane*, *BD_Distance*, *BD_Number of Passengers*, *BD_Mass*, *X*, *Y*, необходимо в дереве концептов «перетащить» атрибуты *Project_Distance*, *Percent of the Deviation_Distance*, *Project_Min Distance*, *Project_Max Distance*, *Project_Number of Passengers*, *Percent of the Deviation_Number of Passengers*, *Project_Min Number of Passengers*, *Project_Max Number of Passengers*, *X*, *Y* на объект *Project_Plane*. Аналогично следует «перетащить» атрибуты *BD_Name of thePlane*, *BD_Distance*, *BD_Number of Passengers*, *X*, *Y* на объект *BD_Plane*.

◆

- ◆ Установите связи между атрибутами *Project_Distance*, *Percent of the Deviation_Distance*, *Project_Min Distance*, *Project_Max Distance*, *Project_Number of Passengers*, *Percent of the Deviation_Number of Passengers*, *Project_Min Number of Passengers*, *Project_Max Number of Passengers*, *X*, *Y* и объектом *Project_Plane*.
- ◆ Установите связи между атрибутом *BD_Name of thePlane*, *BD_Distance*, *BD_Number of Passengers*, *X*, *Y* и объектом *BD_Plane*.

◆

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. 17 показаны свойства объекта *BD_Plane* в закладке *Uses*, причем данный объект имеет атрибуты *BD_Name of the Plane*, *BD_Distance*, *BD_Number of Passengers*, *X*, *Y* (соответствующие связи отображены в закладке *Uses*). В закладке *Used by* атрибута *Project_Distance* отображена связь этого концепта с объектом *Project_Plane* (рис.18).

The screenshot shows the 'OntologyLibrary_TTT' interface. On the left is a tree view of the ontology structure. The 'BD_Plane' object is selected under 'Objects'. The right pane shows the properties of 'BD_Plane' in the 'Uses' tab. The 'Uses' tab lists several attributes: 'BD_Distance', 'BD_Number of Passengers', 'X', 'Y', and 'BD_Name of the Plane'. The 'Used by' tab is also visible but empty.

Parameter	Value
Common	
Description	
ElementType	Object
Name	BD_Plane
Element resource	
ElementID	12
Extra large icon (50x50)	(TBitmap)
Large icon (32x32)	(TBitmap)
Small icon (16x16)	(TBitmap)
Interface options	
Add to palette	<input checked="" type="checkbox"/>
Interface behavior	<interface behavior properties>
Ontology Attributes	
Descendants Uses Used by	
Show link by category Show inherited links	
Attributes	
BD_Distance	
BD_Number of Passengers	
X	
Y	
BD_Name of the Plane	

Рис.17. Атрибуты (свойства) объекта *BD_Plane* в закладке *Uses*

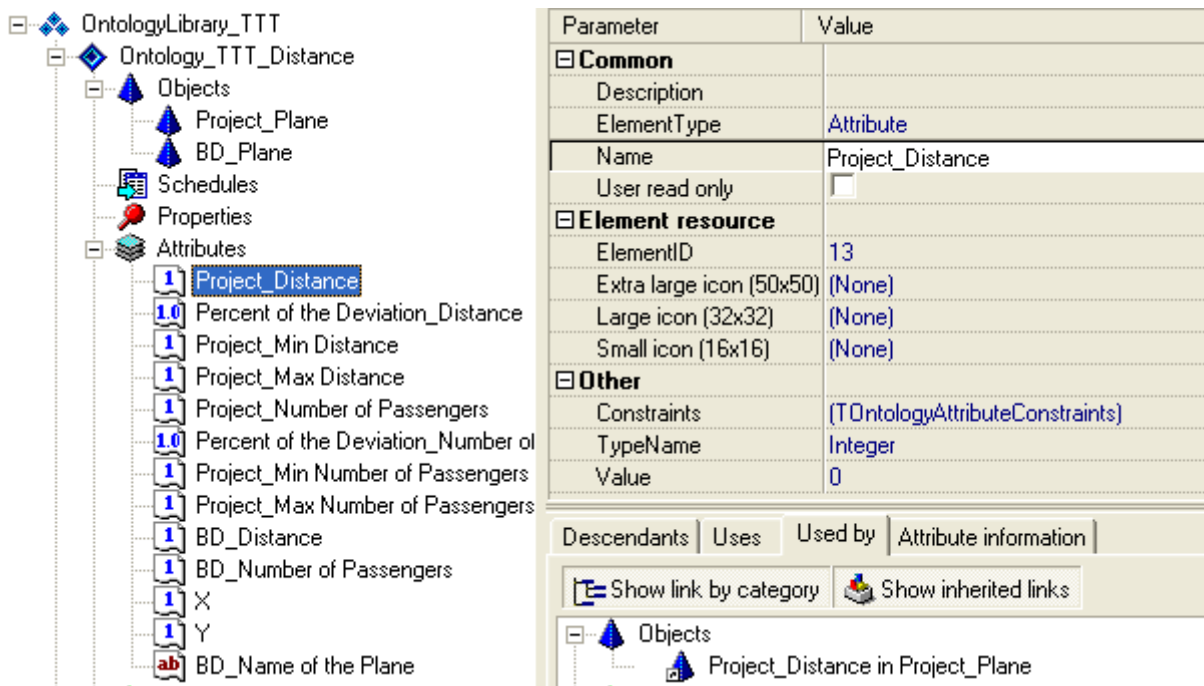


Рис. 18. Связь атрибута с объектом в закладке *Used by*

2.2.2.5 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 -> Arrange* (рис. 19).

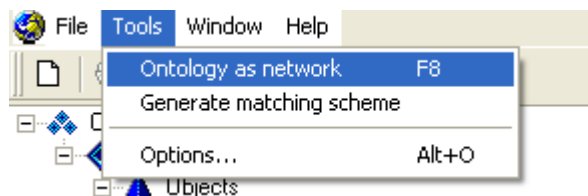


Рис. 19. Вызов представления онтологии в виде семантической сети

В левой части открывшегося окна *Ontology Network* дескриптивная онтология отображается в виде дерева концептов, а в правой части – в виде семантической сети (рис. 20).

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

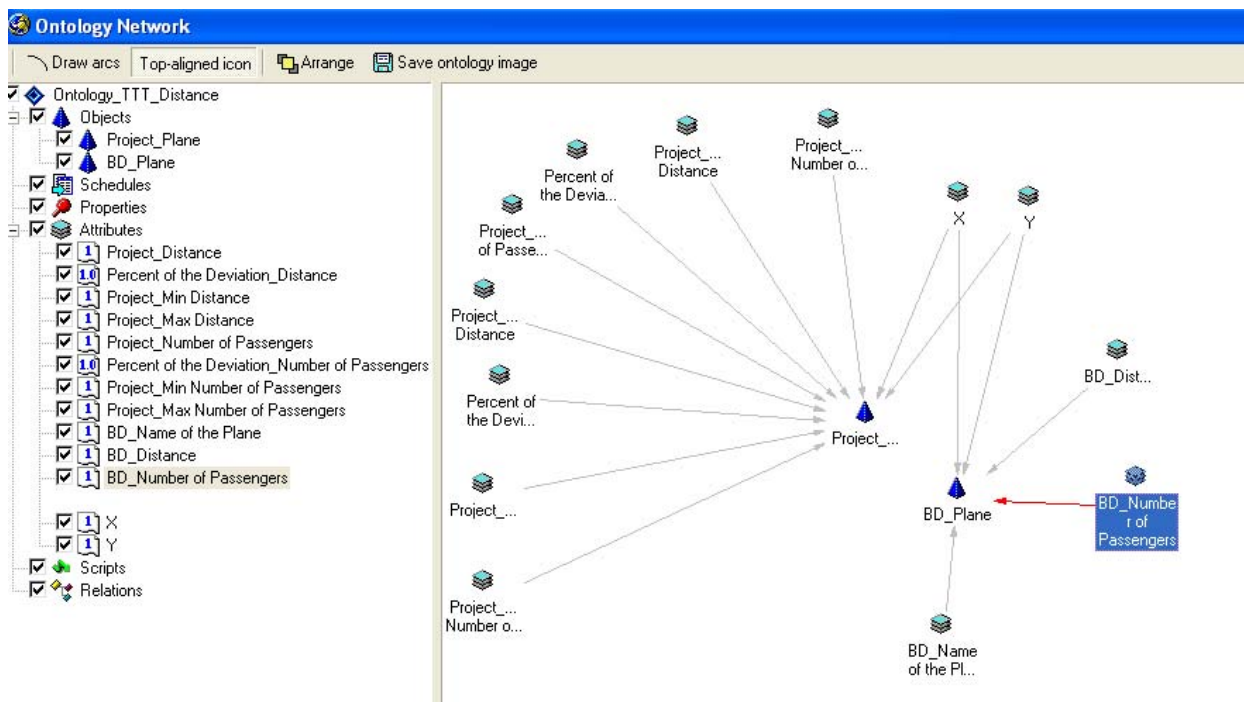




Рис. 20. Представление онтологии в виде семантической сети

2.2.3 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. Например, для того чтобы задать ограничения на диапазон поиска в базе данных прототипа с соответствующими тактико-техническими характеристиками, следует, прежде всего, вычислить минимальное и максимальные допустимые значения дальности полета и количества пассажиров. 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  *Edit script* в закладке *Script body*) (рис. 21).

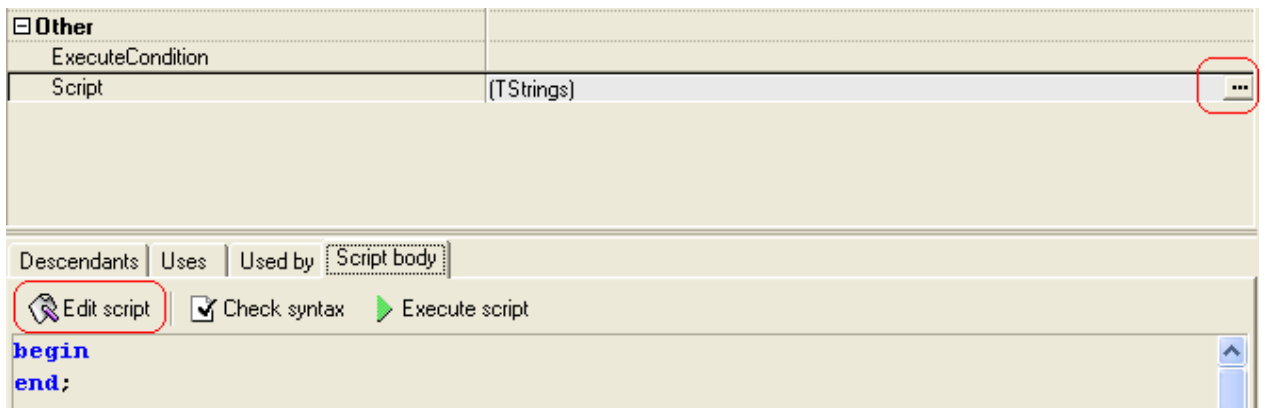


Рис. 21. Закладка *Script body*

Закладка *Script body* предоставляет следующие возможности:

- *Check syntax* – проверить правильность написания скрипта (т.е., выявить наличие синтаксических ошибок),
- *Execute script* – выполнить скрипт и посмотреть вычисленное значение.

Редактор скриптов (*Script editor*) представляет собой текстовый редактор с подсветкой синтаксиса. В левой части редактора скриптов имеется список параметров, содержащий концепты, которые могут быть использованы в скрипте (*Used concepts*) (рис. 22). Чтобы перенести параметр из списка в тело скрипта, необходимо дважды кликнуть на необходимом параметре в списке. Во избежание ошибок, вручную набирать названия параметров не следует.

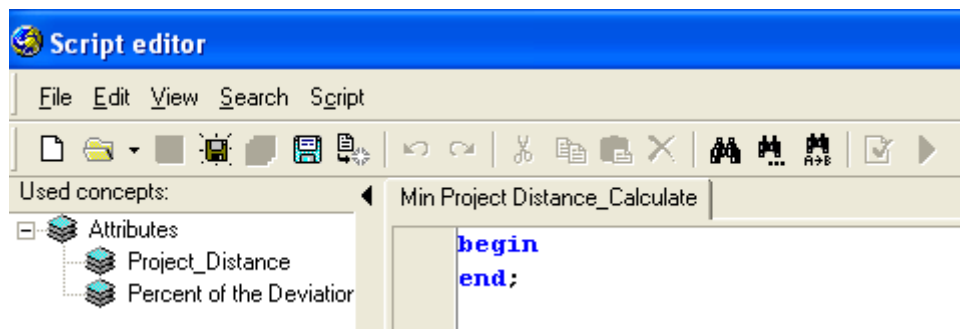


Рис. 22. Редактор скриптов

2.2.3.1 Определение скриптов для вычисления границ поиска самолета-прототипа в базе данных

Чтобы вычислить с помощью скрипта минимальное допустимое значение дальности полета прототипа, необходимо выполнить следующие действия:

- ◆
- ◆ Создайте концепт «скрипт», который будет вычислять минимальное

допустимое значение дальности полета прототипа. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script* (рис. 23).



Рис. 23. Создание концепта «скрипт»

- ◆ Созданный скрипт переименуйте в *Min Project Distance_Calculate* и свяжите его с атрибутом *Project_Min Distance*, т.е. перетащите скрипт на атрибут *Project_Min Distance*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *Min Project Distance_Calculate*, вычисляющего минимальную допустимую дальность полета, являются атрибуты *Project_Distance* и *Percent of the Deviation_Distance* (рис. 24).

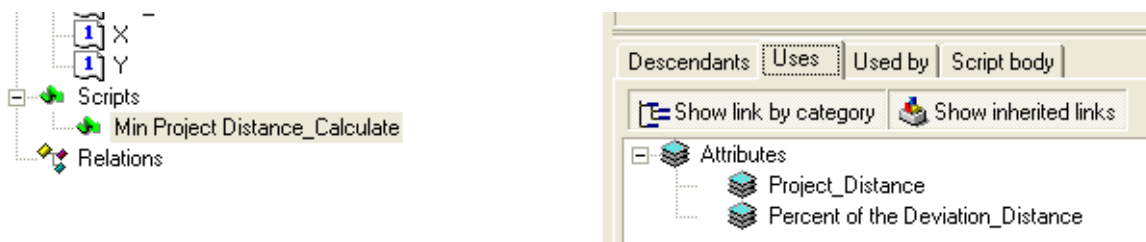
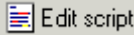


Рис. 24. Атрибуты скрипта *Min Project Distance_Calculate*

- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Min Project Distance_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 (рис. 25).

begin

```
Result := "Project_Distance" - "Percent of the
          Deviation_Distance" * "Project_Distance";
```

end;

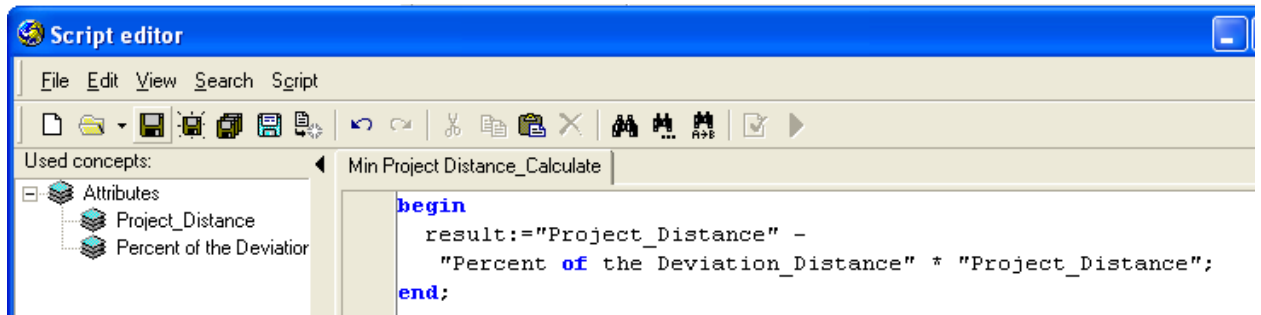



Рис. 25. Атрибуты скрипта *Min Project Distance_Calculate*

- ◆ 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.
- ◆
- ◆ Аналогично создайте концепт «скрипт» *Max Project Distance_Calculate*, который будет вычислять максимальное допустимое значение дальности полета прототипа. Свяжите его с атрибутом *Project_Max Distance*, т.е. перетащите скрипт на атрибут *Project_Max Distance*.

begin

```
Result := "Project_Distance" + "Percent of the
          Deviation_Distance * "Project_Distance";
```

end;

- ◆
- ◆ Аналогично создайте концепт «скрипт» *Min Project Number of Passengers_Calculate*, который будет вычислять минимальное допустимое количество пассажиров прототипа. Свяжите его с атрибутом *Project_Min Number of Passengers*, т.е. перетащите скрипт на атрибут *Project_Min Number of Passengers*. Параметрами скрипта *Min Project Number of Passengers_Calculate*, вычисляющего минимальное допустимое количество пассажиров, являются атрибуты *Project_Number of Passengers* и *Percent of the Deviation_Number of Passengers* (рис. 26).

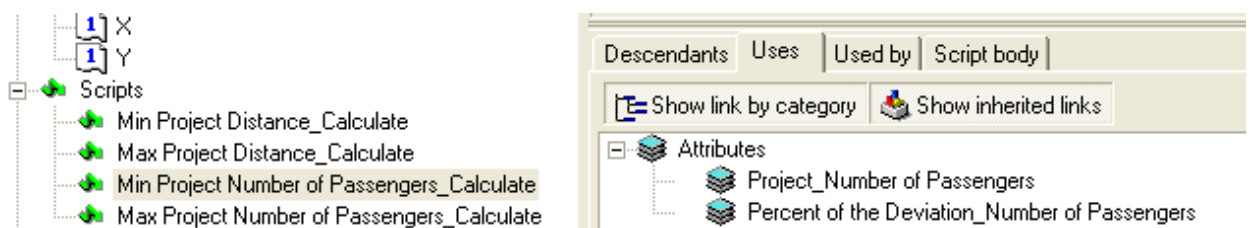


Рис. 26. Атрибуты скрипта *Min Project Number of Passengers_Calculate*

Тело скрипта
begin


```

Result := "Project_Number of Passengers" -
"Percent of the Deviation_Number of Passengers *
"Project_Number of Passengers";
end;

```

- ◆ Аналогично создайте концепт «скрипт» *Max Project Number of Passengers Calculate*, который будет вычислять максимальное допустимое количество пассажиров прототипа.

```

begin
Result := "Project_Number of Passengers" +
"Percent of the Deviation_Number of Passengers *
"Project_Number of Passengers";
end;

```

◆

2.3 ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ МИРА ЗАКАЗОВ И РЕСУРСОВ

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* (рис. 27). 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 (рис. 28). 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 “самолет-прототип в базе данных” выступает в роли ресурса и должен иметь в виртуальном мире только агента ресурса. Нажатие на кнопку *<OK>* подтверждает необходимость создания онтологии виртуального мира.

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. 29).

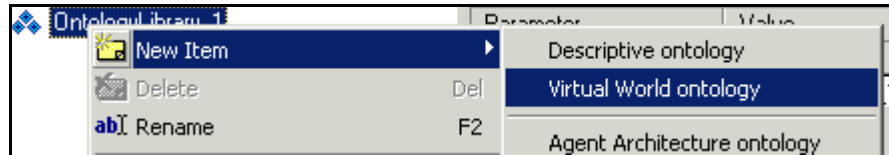


Рис. 27. Создание онтологии мира заказов ресурсов

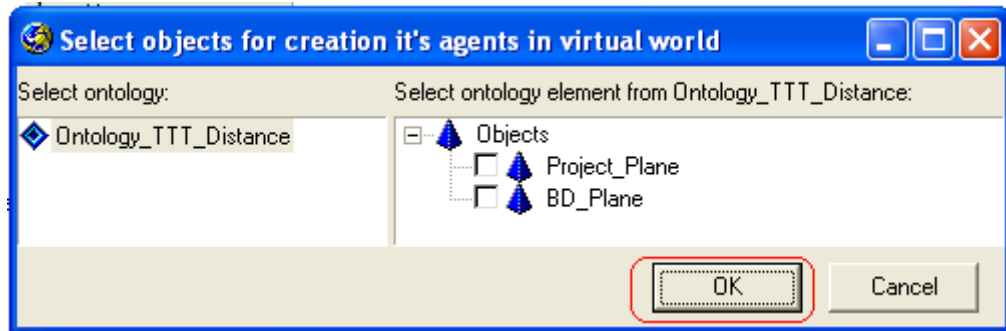


Рис. 28. Выбор объектов, для которых должны создаваться агенты

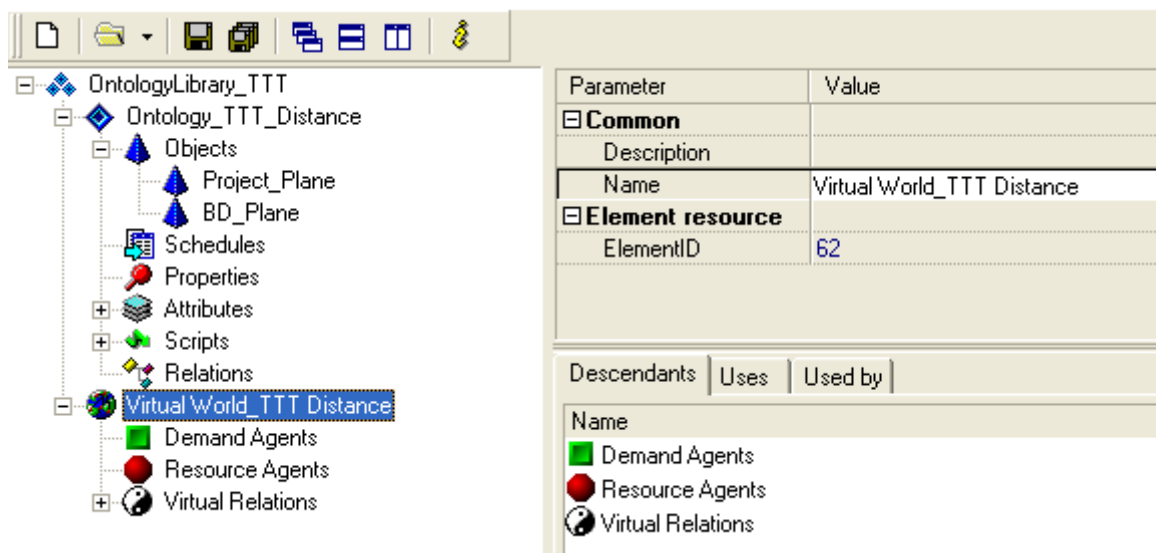


Рис. 29. Категории концептов онтологии мира заказов и ресурсов

- ◆
- ◆ Создайте онтологию мира заказов и ресурсов для предметной области «Тактико-технические требования к самолету» (*New Item -> Virtual World Ontology*). Введите в поле *Name* название онтологии - *Virtual World_TTT_Distance*. Раскройте дерево концептов онтологии виртуального мира (рис. 29).
- ◆

2.3.2 Создание концепта «агент заказа»

Чтобы создать концепт «агент заказа», в архитектуре онтологии виртуального мира необходимо выделить категорию *Demand Agents*, в

контекстном меню выбрать пункт *New Item*. Далее в диалоге создания онтологии виртуального мира выбрать онтологию и концепт «объект» из этой онтологии, с которым должен быть связан создаваемый агент заказа (рис. 30).

Концепт категории «агент заказа» имеет следующие свойства (параметры) (рис. 31):

- *Common* – словесное описание (*Description*), тип (*ElementType*), имя (*Name*) концепта.
- *Compensation properties* – настройки для метода компенсаций (в данной работе не рассматриваются).
- *Element resource* – пиктограмма для представления концепта «агент заказа» в сцене.
- *Virtual world* - флаг *vaoAutoCreate* означает, создается ли виртуальный агент при создании физического объекта (в сцене). Данный флаг должен быть установлен для обычных виртуальных агентов (по умолчанию). Остальные флаги регулируют установки для методов функционирования агентов, которые в данной работе не рассматриваются, устанавливать эти флаги не следует.

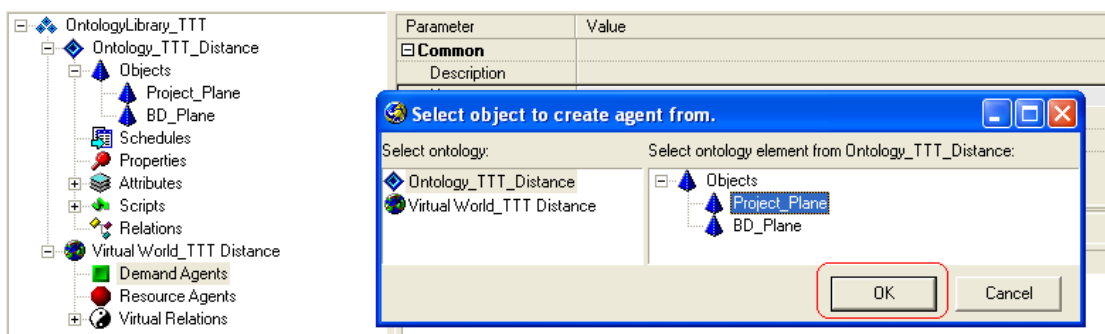


Рис. 30. Создание агента заказа для концепта *Project_Plane*

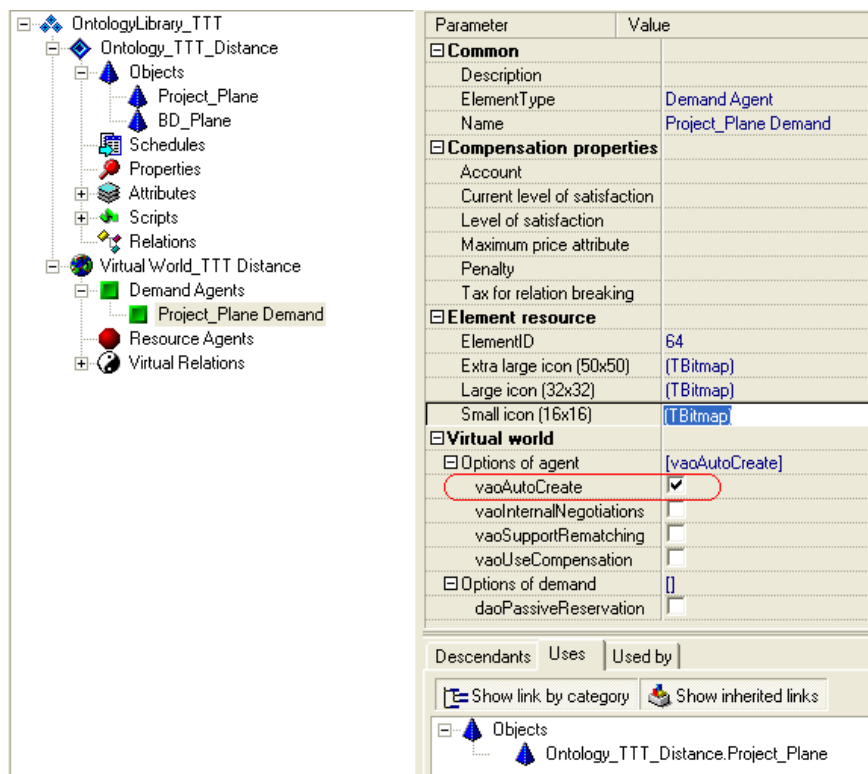


Рис. 31. Свойство концепта «агент заказа»

- ◆
- ◆ 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 *Project_Plane*. Затем нажмите *<OK>* и переименуйте созданный концепт в *Project_Plane Demand*.
- ◆ Назначьте концепту «агент заказа» три вида пиктограмм, с помощью которых данный концепт будет отображаться при работе со сценой в процессе моделирования (см. 2.2.2.1).
- ◆ Установите у концепта флаг *vaoAutoCreate* (должен быть установлен по умолчанию).
- ◆

2.3.3 Создание концепта «агент ресурса»

Чтобы создать концепт «агент ресурса», в архитектуре онтологии виртуального мира необходимо выделить категорию *Resource Agents*, в контекстном меню выбрать пункт *New Item*. Далее в диалоге создания онтологии виртуального мира выбрать онтологию и концепт «объект» из этой онтологии, с которым должен быть связан создаваемый агент ресурса (рис. 32).

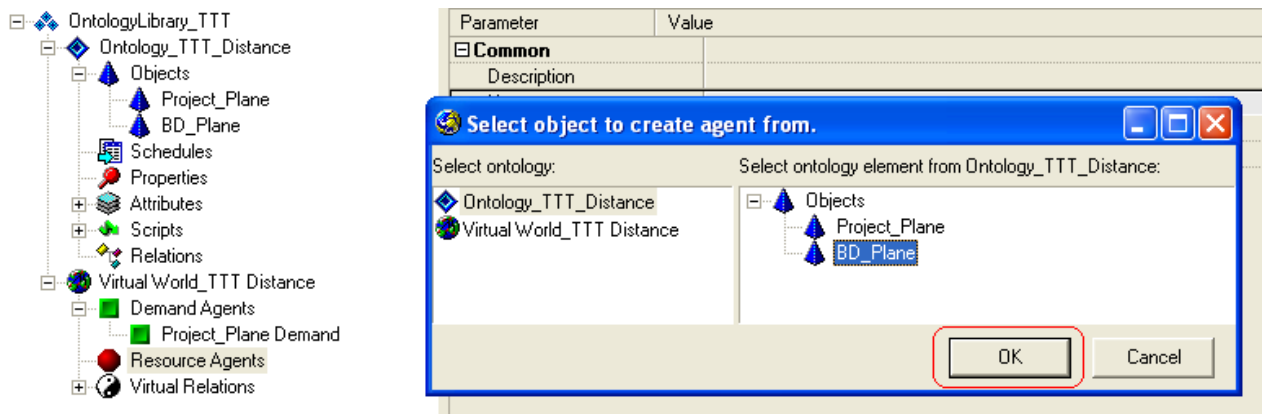


Рис. 32. Создание агента ресурса для концепта *BD_Plane*

Концепт категории «агент ресурса» имеет следующие свойства, аналогичные свойствам концепта категории «агент заказа» (рис. 33):

У концептов «агент ресурса» имеются следующие флаги (параметры *Virtual World*):

- *vaoAutoCreate* (аналогично флагу у концепта «агент заказа»)
- *raoActive* – активность ресурса, т.е., возможность ресурса инициировать матчинг (по умолчанию ресурсы пассивны и не могут инициировать матчинг), флаг *raoActive* сброшен.
- *raoMultiUseable* – возможность мультииспользования ресурса, т.е. совместного использования ресурса несколькими заказами. Если флаг не установлен, то агент будет производить матчинг с заказами, но при наличии одного отношения резервирования не будет давать соглашение на установление второго отношения резервирования.

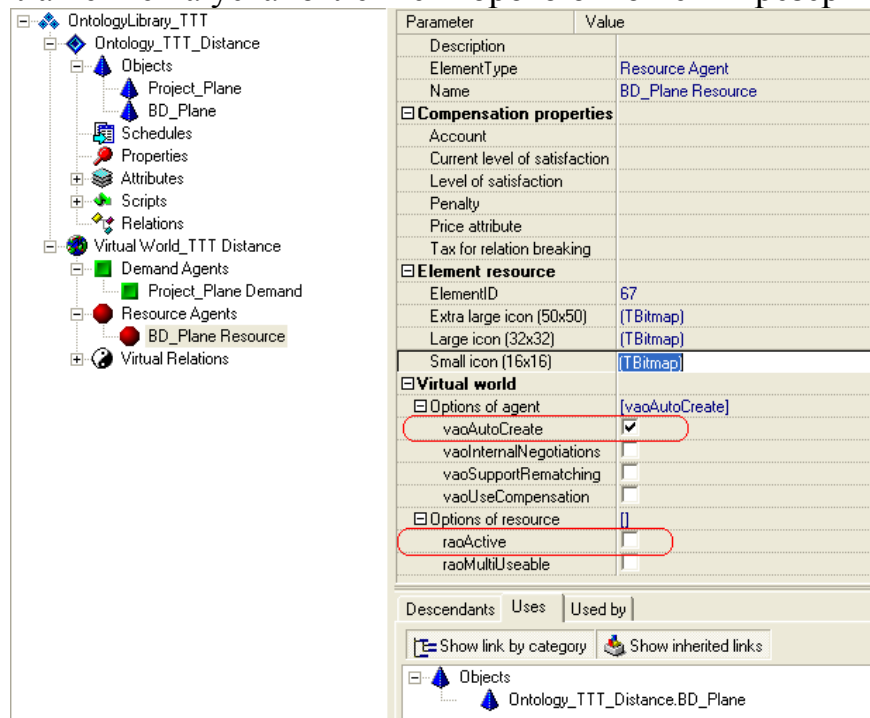


Рис. 33. Свойства концепта «агент ресурса»

- ◆
- ◆ Создайте концепт «агента ресурса» для концепта «самолет-прототип в базе данных»: 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*.
- ◆ Назначьте концепту «агент ресурса» три типа пиктограмм, с помощью которых данный концепт будет отображаться при работе со сценой в процессе моделирования (см. 2.2.2.1).
- ◆ Установите у концепта флаг *vaoAutoCreate*.
- ◆ Не устанавливайте у концепта флаг *raoActive*.
- ◆

2.3.4 Виртуальные отношения: отношение матчинга

Если раскрыть категорию виртуальных отношений, то можно увидеть дерево концептов виртуальных отношений (рис. 34):

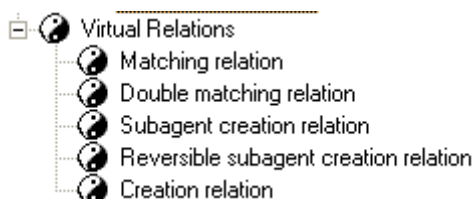


Рис. 34. Виртуальные отношения

В онтологии допускаются отношения двух разновидностей:

- *symmetric* (симметричные)
- *subject-object relations* (субъект - объект).

Отличие их друг от друга состоит в том, что в симметричном отношении все участники имеют равные права, а в отношении «субъект-объект» участники выступают в разных ролях.

В онтологии могут быть созданы виртуальные отношения следующих типов:

- *Matching relation* - отношение одностороннего матчинга с одним активным агентом, который инициирует матчинг;
- *Double matching relation* – отношение двустороннего матчинга с двумя активными агентами, каждый из которых инициирует матчинг;
- *Subagent creation relation* – отношение “агент-зависимый агент (субагент)”;
- *Reversible subagent creation relation* – отношение “агент-разделяемый агент”.

Чтобы связать два концепта отношением, необходимо:

- выделить отношение,
- в контекстном меню выбрать пункт *Establish relation*, после чего появится диалог установки отношения,
- указать концепты, между которыми устанавливается отношение.

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 *Project_Plant Demand* и агента ресурса *BD_Plane 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 *Project_Plane Demand*, а в качестве *Matching object* – *BD_Plane Resource* (рис. 35).



Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Project_Plane Demand.BD_Plane Resource* (рис. 36).

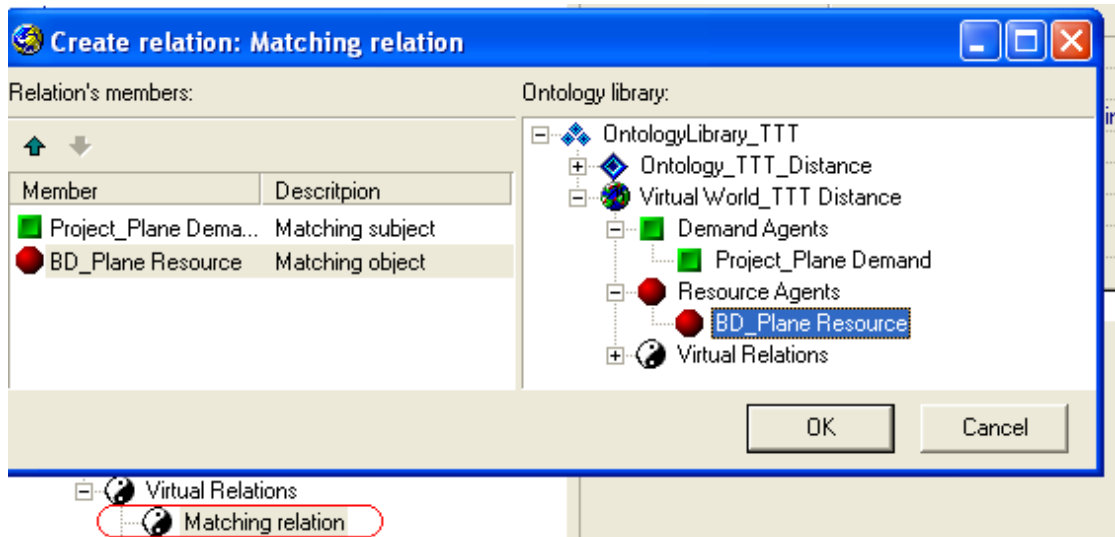


Рис. 35. Связывание отношением матчинга концептов *Project_Plane Demand* и *BD_Plane Resource*

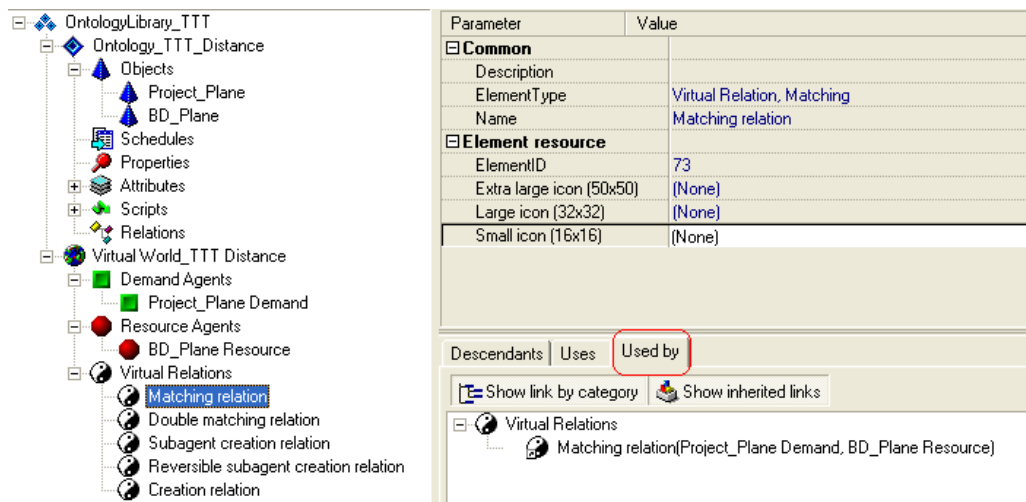


Рис. 36. Свойства концепта «виртуальные отношения: отношение матчинга»

2.3.5 Условия матчинга

Матчинг выполняется при определенных условиях. В конструкторе онтологий имеются следующие условия матчинга:

- знаковое условие (*Signed condition*),
- скриптовое условие (*Scripted condition*).

Знаковое условие матчинга

Знаковое условие матчинга (*Signed condition*) – это условие вида

$\langle SubjectAttribute \rangle \langle \text{Знак} \rangle \langle ObjectAttribute \rangle,$

где $\langle \text{Знак} \rangle$ - это знак сравнения: $>$, $<$, $=$, $<>$, $<=$, $>=$.

SubjectAttribute показывает, какой агент будет проверять данное

условие. При этом множества атрибутов у агентов не должны пересекаться.

Скриптовое условие матчинга

Скриптовое условие матчинга (*Scripted condition*) – это условие, проверяемое с помощью скрипта, который возвращает *True*, если условие матчинга выполняется, и *False* в противном случае.

Скриптовое условие матчинга является наиболее универсальным, с помощью его можно реализовать любое условие матчинга. Однако не следует использовать скриптовые условия там, где можно обойтись знаковыми, так как, как правило, скриптовые условия матчинга проверяются дольше, а в общем случае – не быстрее, и являются менее наглядными при создании онтологии.

Условия матчинга задаются и редактируются в окне редактирования условий матчинга.

- ◆
- ◆ Перейдите в закладку *Used by (Virtual relations -> Matching relation -> Used by)* и выделите *Matching relation (Project_Plane Demand, BD_Plane Resource)*, далее в контекстном меню выберите *Edit virtual relation properties* (рис. 37). При этом откроется окно редактирования условий матчинга (рис. 38).
- ◆

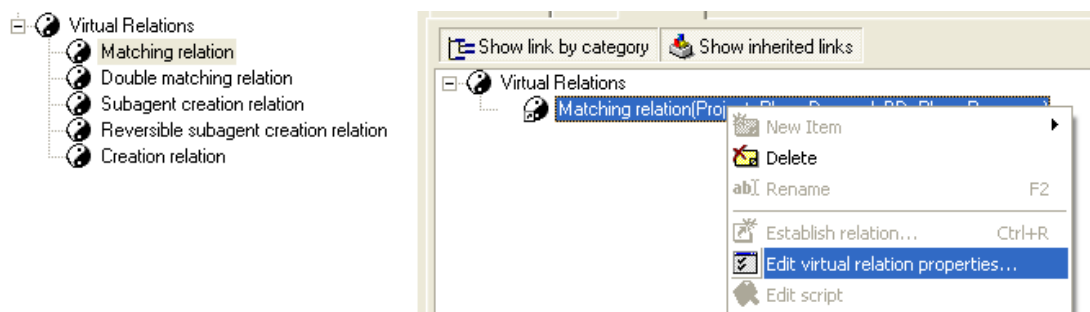


Рис. 37. Редактирование свойств отношения матчинга

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 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).

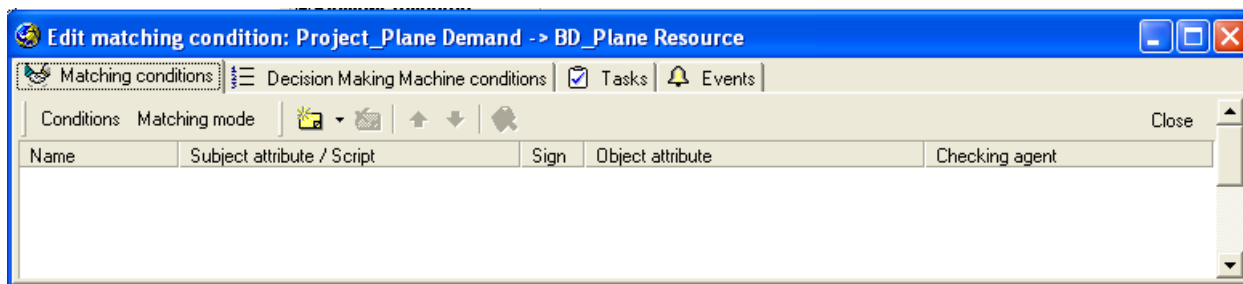




Рис. 38. Окно редактирования условий матчинга

2.3.5.1 Создание условий матчинга, ограничивающих поиск самолетов-прототипов в базе данных


Условия матчинга фактически представляют собой ограничения, согласно которым из всего множества агентов, потенциально участвующих в матчинге, выбираются только те агенты, чьи атрибуты находятся в заданных пределах.

В задаче поиска прототипов самолетов необходимо сравнивать значения атрибутов “дальность полета” и “число пассажиров” прототипа в базе данных с допустимой нижней и верхней границей значения каждого из этих атрибутов при проектировании.

Создать условие матчинга можно нажатием кнопки , для удаления условия матчинга используется кнопка .

Рассмотрим создание условия минимально допустимой дальности полета (рис. 39).

◆

- ◆ Создайте условие минимальной дальности полета. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:

- SubjectAttribute = ‘BD_Plane Resource.BD_Distance’;
- ObjectAttribute=‘Project_Plane Demand.Project_Min Distance’;
- Sign = ‘>=’.

Значения *SubjectAttribute*, *ObjectAttribute*, *Sign* выбираются из выпадающих меню, которые открываются в соответствующих полях в нижней части окна *Edit matching condition*.

Данное условие означает, что агент проектируемого самолета при матчинге будет выбирать только те самолеты-прототипы, у которых дальность полета не меньше требуемой с учетом процента отклонения,

допустимого при выборе.

- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆

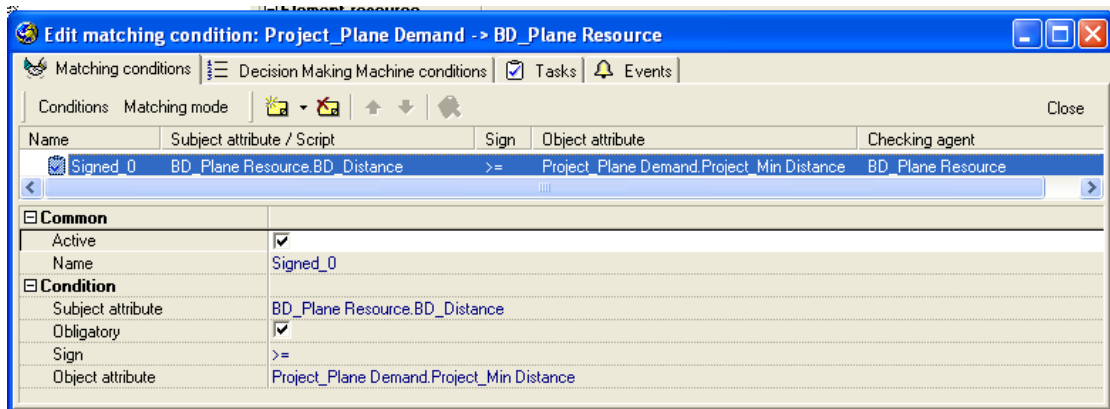






Рис. 39. Редактирование условия матчинга

- ◆
- ◆ Создайте условие максимально допустимой дальности полета. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
 - SubjectAttribute = 'BD_Plane Resource.BD_Distance';
 - ObjectAttribute='Project_Plane Demand.Project_Max Distance';
 - Sign = '<='.
- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆
- ◆ Создайте условие минимально допустимого количества пассажиров. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
 - SubjectAttribute = 'BD_Plane Resource.BD_Number of Passengers';
 - ObjectAttribute='Project_Plane Demand.Project_Min Number of Passengers';
 - Sign = '>='.
- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆
- ◆ Создайте условие максимально допустимого количества пассажиров. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
 - SubjectAttribute = 'BD_Plane Resource.BD_Number of Passengers';
 - ObjectAttribute='Project_Plane Demand.Project_Max Number of Passengers';

- Sign = '<='.
- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆

Все активные условия учитываются в процессе матчинга. Условия можно временно отключать, сбрасывая флажки *Active* и *Obligatory*. Выделенное ограничение можно переместить вниз или вверх в списке ограничений с помощью стрелок .

Условия матчинга для задачи поиска прототипа по дальности полета и количеству пассажиров приведены на рис. 40.

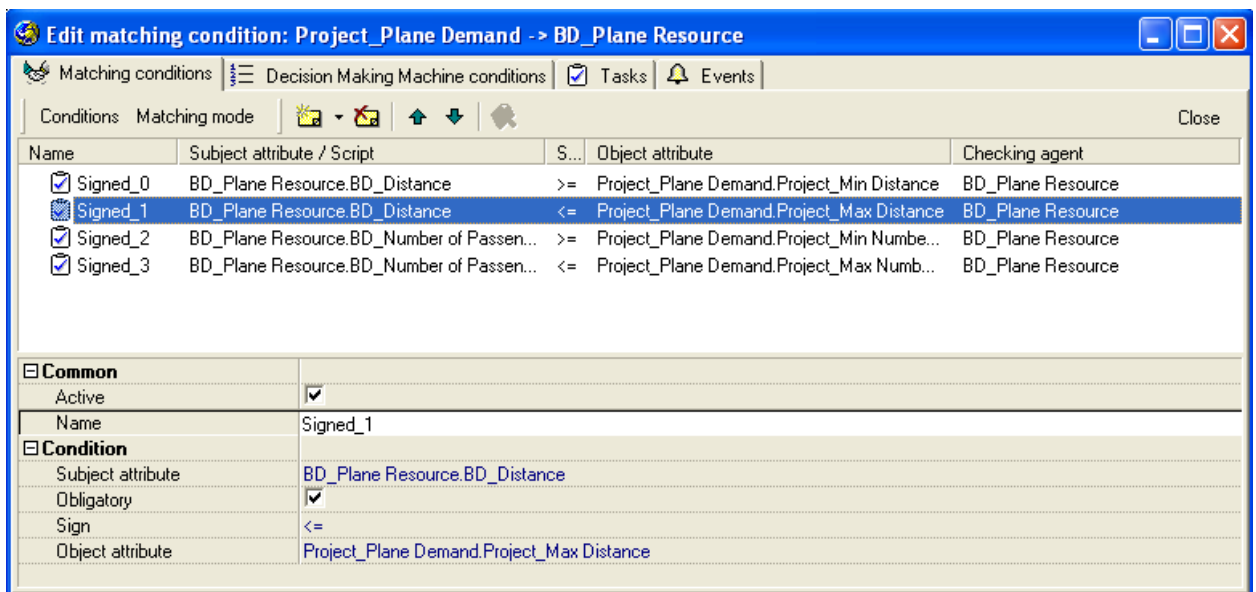



Рис. 40. Условия матчинга для задачи поиска прототипа по дальности полета и количеству пассажиров

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 (рис. 41), определяющий «значимость» данного условия.

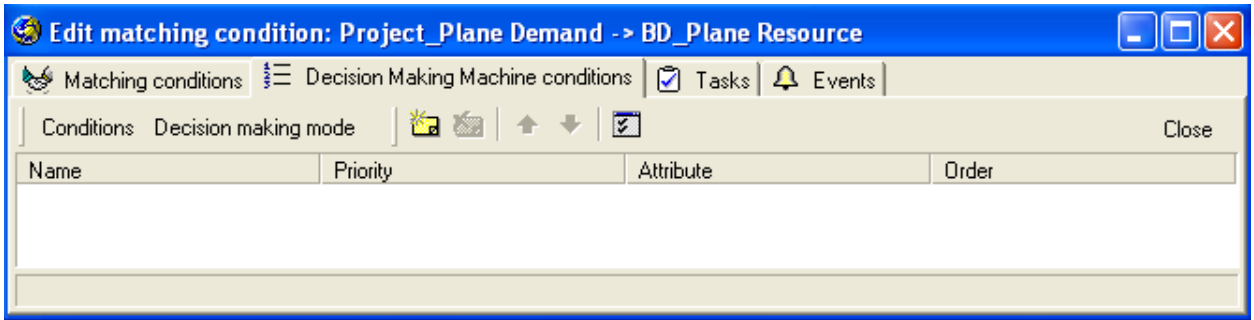



Рис. 41. Окно редактирования условий принятия решения

2.3.6.1 Создание условия принятия решения - максимизация дальности полета самолета-прототипа



- ◆
- ◆ Создайте условие принятия решений для матчинга *Project_Plane Demand* -> *BD_Plane Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия:

- Attribute = 'BD_Plane Resource.BD_Distance';
- Order = 'Max';
- Weight = '100'.

Условие означает, что агент проектируемого самолета при матчинге более, чем с одним агентом самолета-прототипа, будет выбирать самолет-прототип с максимальной дальностью полета.

- ◆ Активизируйте условие принятия решения, выставляя флажок *Active*.

◆ Все активные критерии учитываются в процессе поиска решения. Условия можно временно отключать, сбрасывая флажок *Active*.

Если критериев принятия решения несколько, выделенный критерий можно переместить вниз или вверх в списке критериев с помощью стрелок . Если задается несколько критериев принятия решения, дополнительно указывается метод сортировки альтернатив, который можно выбрать в меню, открываемом при нажатии на кнопку  в закладке *Decision Making Machine conditions* (в данной работе не рассматривается).

Условие принятия решения в матчинге для поиска прототипа по дальности полета и количеству пассажиров приведены на рис. 42.

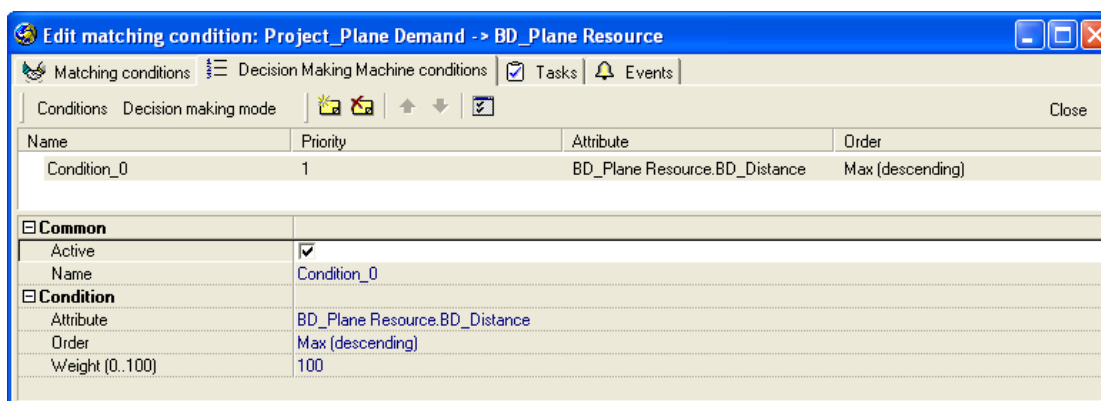


Рис. 42 – Задание условия принятия решений матчинга *Project_Plane Demand – BD_Plane Resource*

2.3.7 Параметры представления экземпляра объекта в сцене

Поведение экземпляра концепта «объект» (агента) в сцене определяет Группа свойств концепта «объект» *Interface behaviour*, которая имеет следующие параметры (рис. 43):

- *Agent nexus* – уникальный идентификатор экземпляра концепта в сцене. Значение данного параметра определяется в момент создания агента. Если в процессе матчинга необходимо оперировать уникальными идентификаторами, то следует в поле *Agent nexus* указать соответствующий атрибут (*Agent reference attribute*), и тогда в момент создания агента атрибуту будет присвоено значение идентификатора (ссылка).
- *Agent name* – имя агента. С данным параметром должен быть связан строковый атрибут (*String attribute*).
- *Usage level* – величина, выражающая степень использования агента, соответствующего данному объекту, в процентах. С данным параметром должен быть связан атрибут «вещественное число» (*Float attribute*).
- *Level of satisfactory* – степень удовлетворения агента. Может принимать значения от 0 до 100. В интерфейсе данному параметру соответствует набор иконок около агентов, с помощью которых отображается данная степень. С этим параметром должен быть связан атрибут «вещественное число» (*Float attribute*).

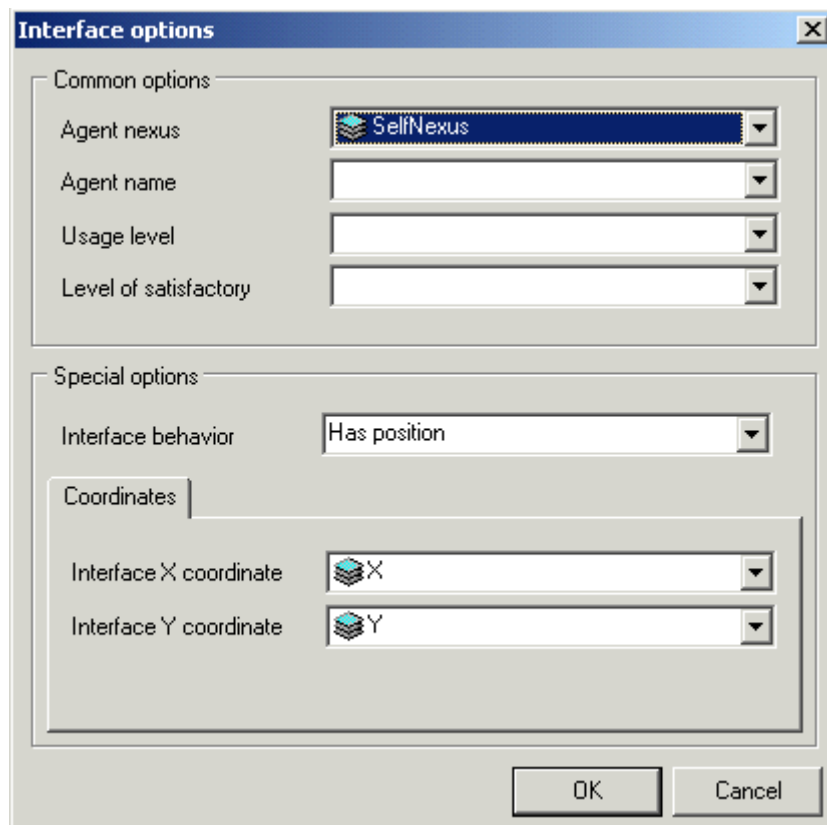


Рис. 43. Редактор свойств *Interface behaviour*

- *Interface behaviour* – в зависимости от его значений могут задаваться следующие параметры:
 - *Has position* указывает на то, что объект имеет позицию на карте в интерфейсе.
 - *Interface X coordinate* – координата объекта по оси *X* на карте в интерфейсе. С данным параметром должен быть связан атрибут «целое число» (*Integer attribute*).
 - *Interface Y coordinate* – координата объекта по оси *Y* на карте в интерфейсе. С данным параметром должен быть связан атрибут «целое число» (*Integer attribute*).
 - *Docked* указывает на то, что объект «привязан» к другому объекту.
 - *Dock object* – уникальный идентификатор агента, к которому «привязан» данный объект. Заметим, что объект может «привязываться» не только к объектам на карте, но и к коннекторам (линиям). С данным параметром должен быть связан атрибут «ссылка на агента» (*Agent reference attribute*).
 - *Position on connector* – показывает расположение объекта на коннекторе. Расположение задается в процентах от длины коннектора (линии) и отсчитывается от начала коннектора. Таки образом значение 0% показывает, что объект находится в начале линии, 25% - прошёл четверть пути, 50% - в середине и т.д. С данным параметром должен быть связан атрибут

«вещественное число» (*Float attribute*).

- *Connector* указывает на то, что объект является коннектором (линией, соединяющей объекты в интерфейсе).
 - *Connector source* – уникальный идентификатор агента, являющегося началом коннектора (линии).
 - *Connector destination* – уникальный идентификатор агента, являющегося окончанием коннектора (линии).
 - *Connector length* – длина линии, отображающей коннектор. С данным параметром должен быть связан атрибут «вещественное число» (*Float attribute*).

Чтобы задать (изменить) свойство *Interface behaviour*, необходимо выделить нужный концепт «объект», далее в редакторе свойств концепта в параметре *Interface options* -> *Interface behaviour* вызвать диалог редактирования параметров *Interface behaviour* (рис. 44).

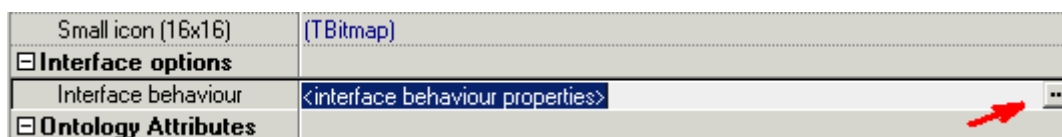


Рис. 44. Задание свойств *Interface behaviour*

В появившемся диалоговом окне следует установить необходимые параметры посредством выбора нужных атрибутов из выпадающих списков. Атрибуты выбираются из общего множества установленных атрибутов редактируемого концепта «объект», тип которых является корректным для выбранного параметра. Чтобы разорвать ассоциацию атрибута с параметром свойства *Interface behaviour*, необходимо нажать клавишу . Для применения установок необходимо нажать клавишу <OK>.

- ◆
- ◆ Свяжите атрибуты *X*, *Y* с позицией агента объекта *Project_Plane* в сцене. Для этого в редакторе свойств объекта *Project_Plane* 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 *BD_Plane* в сцене. Для этого в редакторе свойств объекта *BD_Plane* 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>.
- ◆

2.3.8 Сохранение онтологий предметной области “Тактико-технические требования к

проектируемому самолету”

Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Тактико-технические требования к проектируемому самолету” приведено на рис. 45.

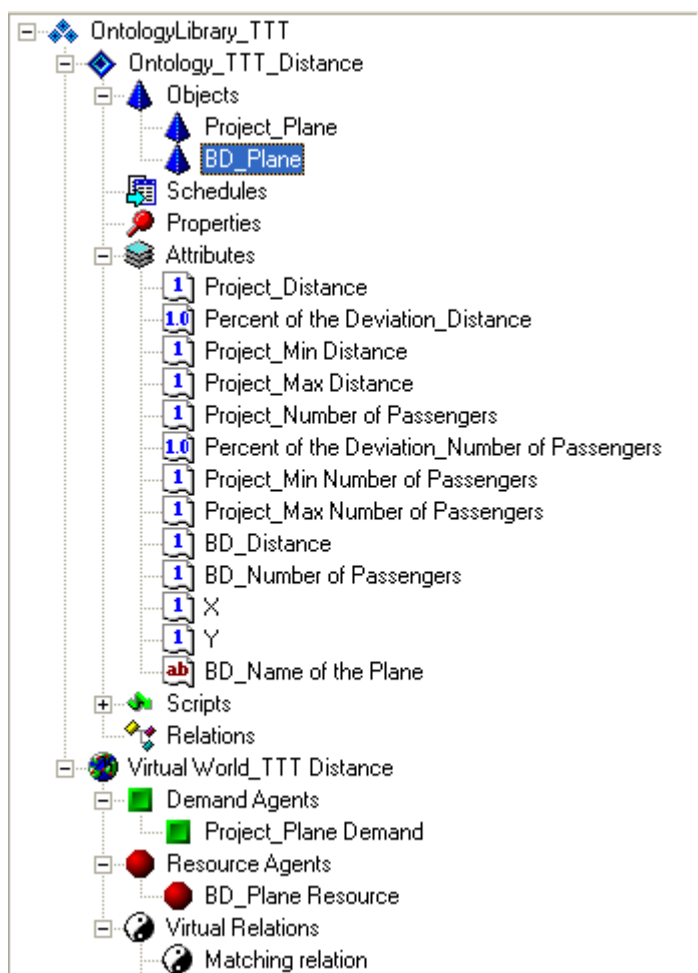



Рис. 45. Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “ТТТ к проектируемому самолету”

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

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. 46).

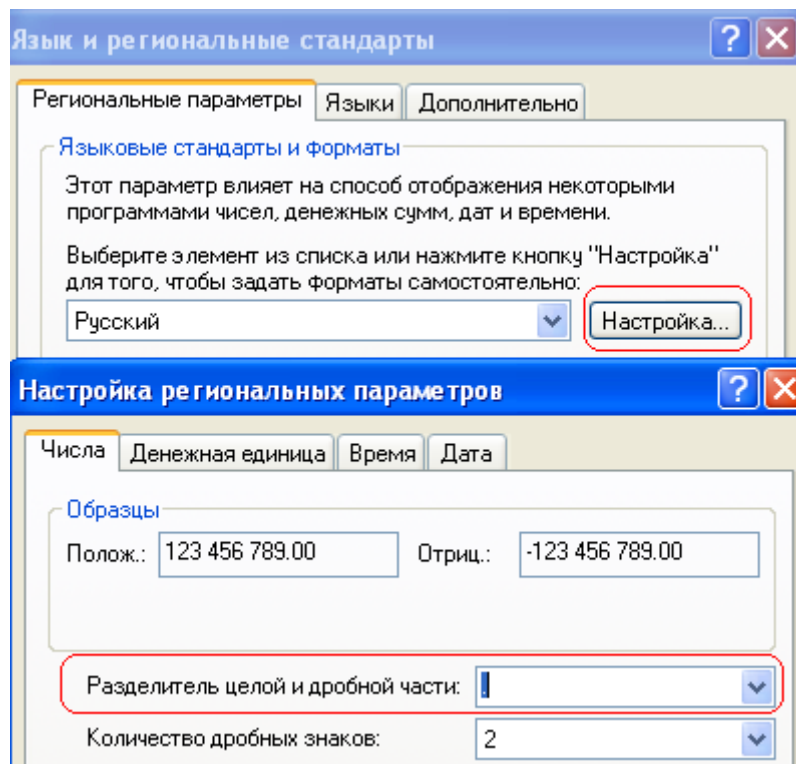



Рис. 46. Настройки компьютера

- ◆ Вызовите исполняющую систему, запуская программу  `Unifntf.exe`, находящуюся в папке `OntConsUniIntf`.
- ◆ Если при загрузке исполняющей системы было открыто окно системного лога, о чем свидетельствует наименование *Magenta Unified Interface – [Central Log]* в поле названия окна исполняющей системы, закройте или сверните окно *Central Log* (рис. 47). После свертывания или закрытия окна системного лога в поле названия окна остается наименование *Magenta Unified Interface*.

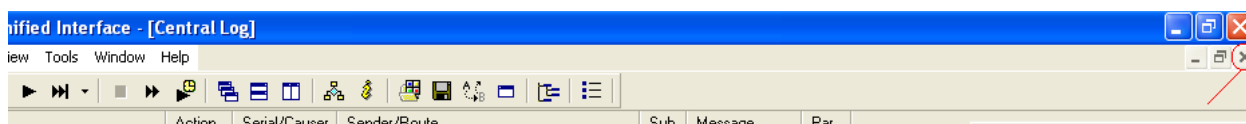



Рис. 47. Закрытие окна лога сообщений

Для создания новой онтологической сцены необходимо воспользоваться пунктом меню *File* → *New scene*, сочетанием клавиш `<Ctrl>+<N>` или выбрать кнопку  на рабочей панели. На экране появляется окно диалога для создания новой сцены (рис. 48). Далее необходимо нажать кнопку *Load ontology* и выбрать созданную ранее онтологию.

◆

- ◆ Создайте новую онтологическую сцену (*File* → *New scene* -> *Load ontology*, выберите онтологию *TTT_Distance ontology.ocl*).

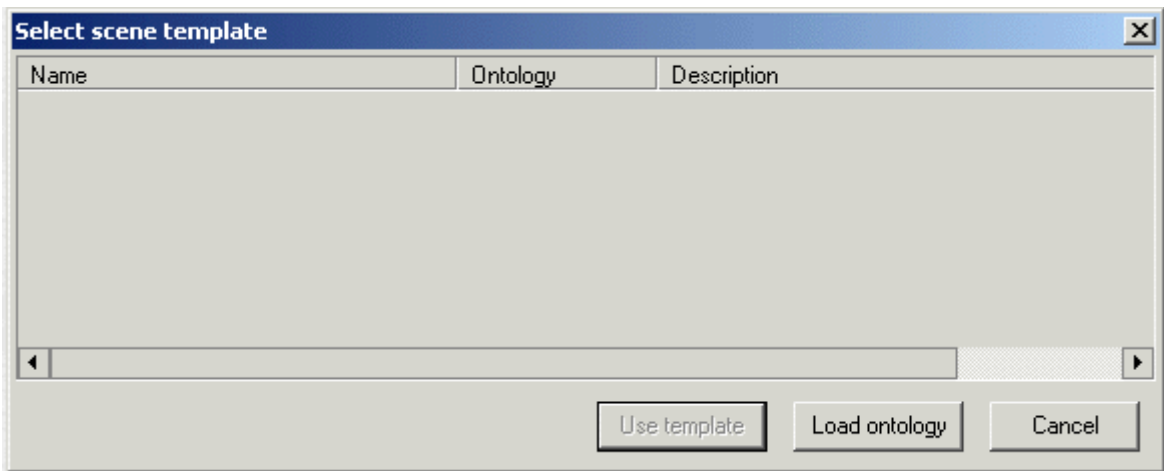


Рис. 48. Создание новой онтологической сцены

- ◆ В окне физического мира создайте одного агента проектируемого самолета и девять агентов самолетов-прототипов в базе данных (рис. 49). Процентные соотношения следует задавать в виде вещественных чисел, принимая 100% за единицу. Допустимое отклонение при подборе самолета-прототипа по дальности полета составляет 20% (значение атрибута *Percent of the Deviation_Distance* = 0.2 у агента проектируемого самолета), а допустимое отклонение при подборе самолета-прототипа по количеству пассажиров составляет 10% (значение атрибута *Percent of the Deviation_Number of the Passengers* = 0.1 у агента проектируемого самолета). С помощью инспектора агентов установите следующие значения атрибутов для агентов:

<i>Имя агента</i>	<i>Тип самолета</i>	<i>Дальность полета, км</i>	<i>Количество пассажиров</i>	<i>Взлетная масса, кг</i>
Project_Plane Demand_1		2000	100	
BD_Plane Resource_1	Бритиш Аэро-спейс 146 - 300	2000	122	44230
BD_Plane Resource_2	Бритиш Аэро-спейс 146 - 100	1730	93	38010
BD_Plane Resource_3	Бритиш Аэро-спейс 146 - 200	2180	109	42180
BD_Plane Resource_4	Фоккер 70	2000	79	36740
BD_Plane Resource_5	Фоккер 100	2390	107	43090
BD_Plane Resource_6	Аеро Интернешнл	2260	100	46000

	Ридженд RJ100			
BD_Plane Resource_7	ТУ - 134Б	2020	96	47600
BD_Plane Resource_8	ТУ – 334 - 100	2000	110	46100
BD_Plane Resource_9	ЯК – 42Д	2150	120	56500

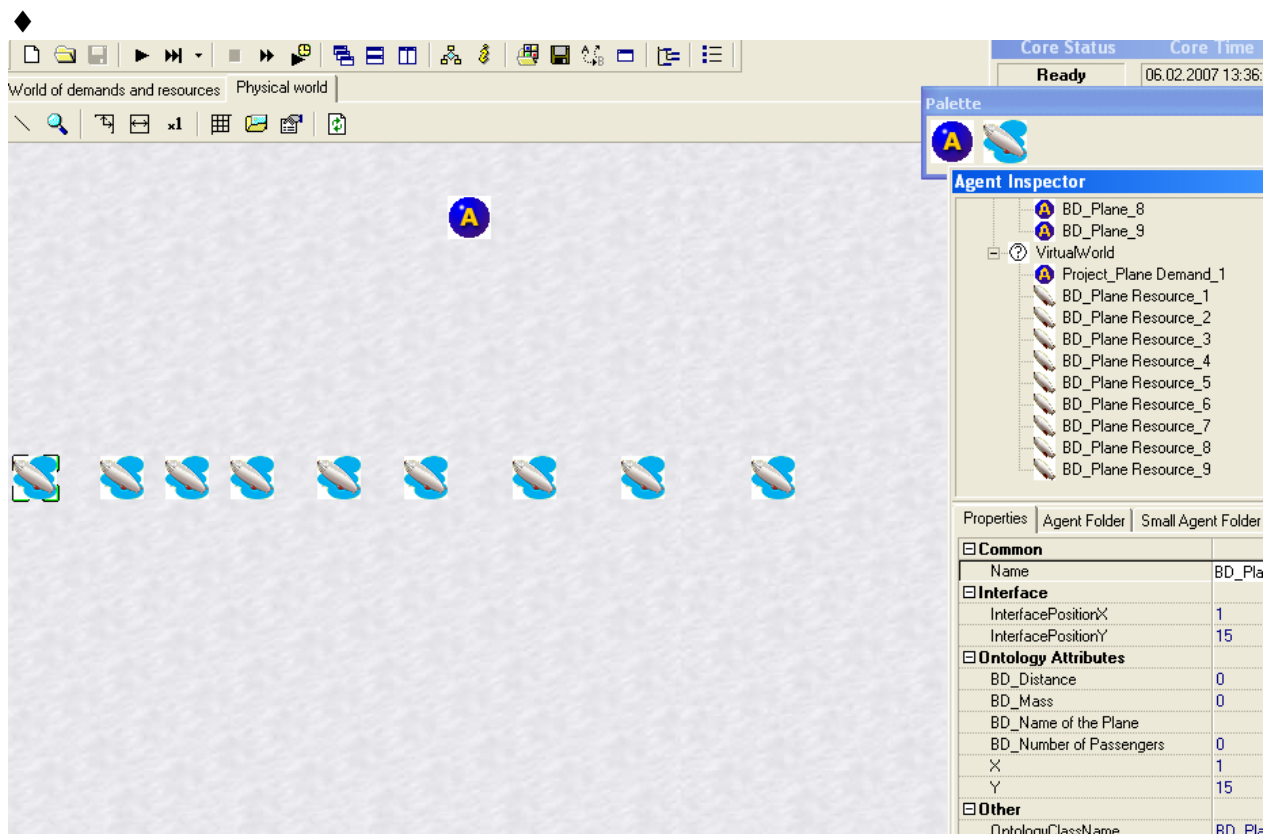




Рис. 49. Создание онтологической сцены предметной области «ТТТ к проектируемому самолету»

2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА


2.5.1 Запуск сцены на моделирование

После создания сцены необходимо выполнить ее моделирование, запуская тем самым процесс матчинга агентов. Процесс матчинга запускается на исполнение нажатием на кнопку  панели инструментов, или на клавишу <F9>, или выбором пункта *Run* → *Run* главного меню. (Моделирование в пошаговом режиме выполняется при нажатии клавиши ). После этого в окне виртуального мира (*Worlds of demands and resources*) можно наблюдать результаты матчинга (пунктирная синяя линия показывает, что два объекта согласились на матчинг, сплошная розовая линия показывает, что заказ зарезервировал ресурс).

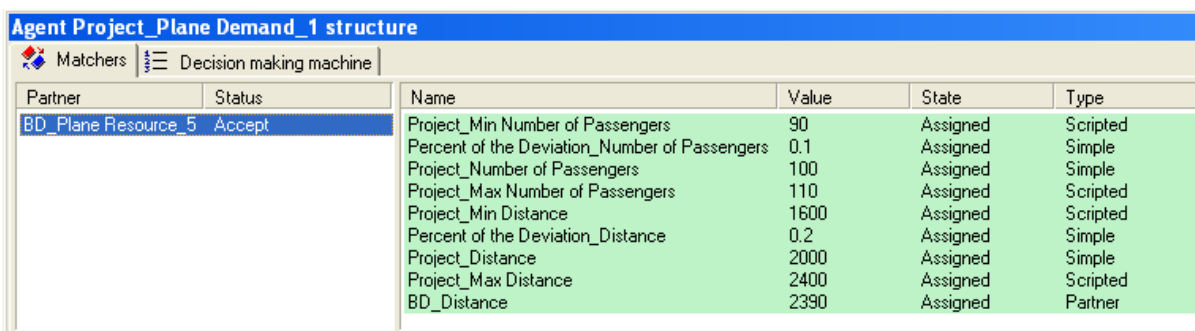
◆

◆ 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.
- ◆

Процесс матчинга начинается с того, что агент проекта (*Project Plane Demand_1*) проверяет значения атрибутов агентов самолетов-прототипов и выбирает агентов, у которых значения атрибутов соответствуют ограничениям на минимально и максимально допустимые значения дальности полета и количества пассажиров. Дальность полета должна быть в пределах 1600 – 2400 км, а количество пассажиров – в пределах 90 – 110 человек. Ограничениям по дальности полета соответствуют все прототипы, но самолеты-прототипы 1, 4, 9 не соответствуют ограничениям по количеству пассажиров, поэтому агент проектируемого самолета не соглашается на матчинг с ними (синие пунктирные стрелки в результатах матчинга отсутствуют). Далее агент проекта строит таблицу принятия решений, в которой размещает прототипы по убыванию дальности полета, и выбирает из таблицы лучший самолет-прототип *BD_Plane_Resource_5* по критерию максимальной дальности полета. Агент проекта резервирует агента прототипа *BD_Plane_Resource_5* (Fokker 100), который удовлетворяет его требованиям: *Project Plane Demand_1* – *BD_Plane_Resource_5* (сплошная малиновая линия в результатах матчинга). По двойному клику левой кнопкой мыши на пиктограмме агента 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 таблице указывается список ресурсов (партнеров), т.е., агентов самолетов-прототипов, которые зарезервированы агентом-проектом (рис. 50). 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
BD_Plane_Resource_5	Accept	Project_Min Number of Passengers	90	Assigned	Scripted
		Percent of the Deviation_Number of Passengers	0.1	Assigned	Simple
		Project_Number of Passengers	100	Assigned	Simple
		Project_Max Number of Passengers	110	Assigned	Scripted
		Project_Min Distance	1600	Assigned	Scripted
		Percent of the Deviation_Distance	0.2	Assigned	Simple
		Project_Distance	2000	Assigned	Simple
		Project_Max Distance	2400	Assigned	Scripted
		BD_Distance	2390	Assigned	Partner

Рис. 50. Структура агента проекта *Project_Plane Demand_1*

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 51), в которой перечислены все агенты самолетов-прототипов, с которыми возможен матчинг в соответствии с критерием принятия решения (минимальной взлетной массой).

Agent	BD_Distance
BD_Plane Resource_5	2390
BD_Plane Resource_6	2260
BD_Plane Resource_3	2180
BD_Plane Resource_7	2020
BD_Plane Resource_8	2000
BD_Plane Resource_2	1730

Рис. 51. Таблица принятия решений агента *Project_Plane Demand_1*

Аналогично можно посмотреть структуру агента-ресурса (рис. 52). В правой части таблицы указаны атрибуты агента-ресурса (*Simple*), а также атрибуты агента-проекта, которые использовались в процессе принятия решения о резервировании (*Partner*). Отображаются только атрибуты, используемые в процессе матчинга (название самолета-прототипа не отображается).

Partner	Status	Name	Value	State	Type
Project_Plane Demand_1	Passive	BD_Number of Passengers	107	Assigned	Simple
		Project_Min Number of Passengers	90	Assigned	Partner
		Project_Max Number of Passengers	110	Assigned	Partner
		BD_Distance	2390	Assigned	Simple
		Project_Min Distance	1600	Assigned	Partner
		Project_Max Distance	2400	Assigned	Partner

Рис. 52. Структура агента ресурса *BD_Plane Resource_2*

Окончательные результаты матчинга можно видеть на рисунке 53. В результате матчинга была выполнена следующая операция резервирования:

- *Project_Plane Demand_1 – BD_Plane Resource_5.*

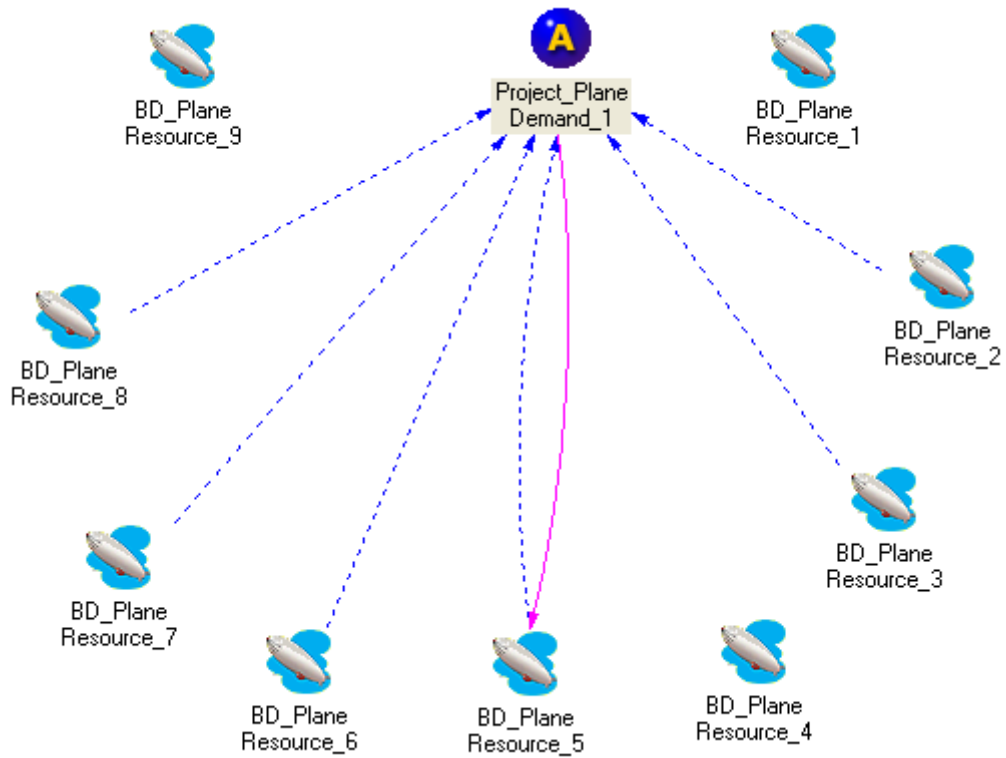



Рис. 53. Результаты матчинга

2.5.2 Сохранение сцены виртуального мира

Для сохранения сцены используются средства онтологического сохранения (рис. 54): кнопка  или последовательность команд *Tools - > Save ontology scene...*

- ◆
- ◆ Сохраните сцену под именем *Scene_TTT_Distance ontology_1*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*.
- ◆ Завершите работу с исполняющей системой (*File -> Close*).
- ◆

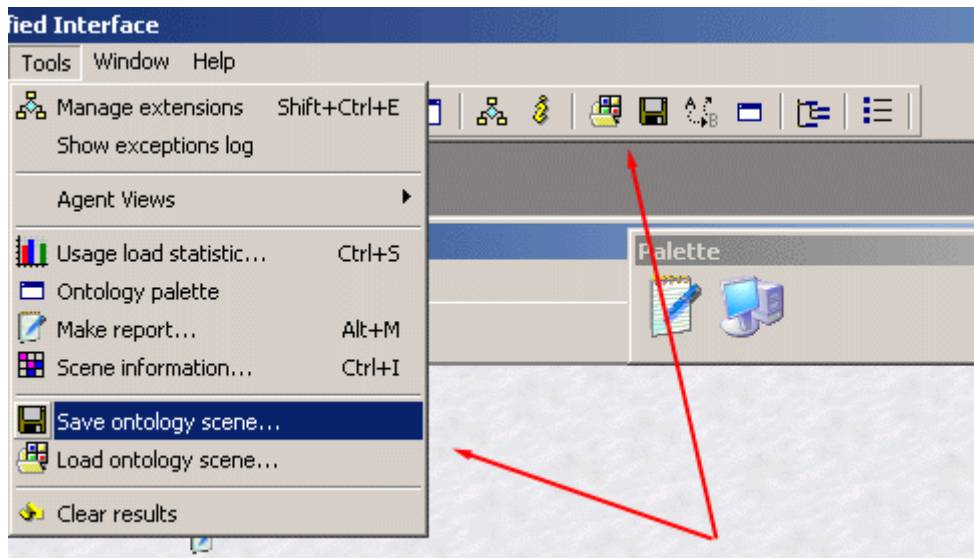




Рис. 54. Средства онтологического сохранения сцены

2.5.3. Загрузка ранее созданной сцены виртуального мира

Для загрузки ранее созданной сцены используются средства онтологической загрузки (рис. 54): кнопка  или последовательность команд *Tools* - > *Load ontology scene...*


- ◆
- ◆ Загрузите сцену *Scene_TTT_Distance ontology_1.osf*.
- ◆

2.5.4 Изучение возможностей системного лога

- ◆
- ◆ Находясь в окне виртуального мира (вкладка *World of demands and resources*), выполните команды *Tools* -> *Clear results*. При этом все результаты предыдущего сеанса моделирования будут очищены, и все связи между агентами, ранее установленные в процессе матчинга, будут разрушены.
- ◆ Выполните команды *View* → *Show Log* (, <Ctrl> + <L>), чтобы загрузить окно сообщений.
- ◆ Находясь в окне системного лога, вызовите контекстное меню -> *Filter*, установите следующие фильтры окна системного лога (рис. 55):
- ◆


<input checked="" type="checkbox"/>	000100	0	GetAgent
<input checked="" type="checkbox"/>	000101	0	SendAgent
<input type="checkbox"/>	000102	0	Question
<input checked="" type="checkbox"/>	000103	0	BeginMatching
<input type="checkbox"/>	000104	0	Answer
<input checked="" type="checkbox"/>	000105	0	Finished
<input checked="" type="checkbox"/>	000106	0	Reserve

Рис. 55. Установка фильтров окна системного лога


- *GetAgent* – получить агента,
 - *SendAgent* – послать агента,
 - *BeginMatching* – начало матчинга между агентами,
 - *Finished* – окончание матчинга между агентами,
 - *Reserve* – резервирование.
- ◆ При помощи кнопки  запустите сцену на моделирование.
 - ◆ Наблюдайте в окне системного лога обмен сообщениями между агентами в процессе матчинга (рис. 56). Различными цветами выделены сообщения о начале и окончании матчинга между агентами, а также о резервировании агентом заказа агента ресурса.

VirtualWorld.Project_Plane Demand_1	Accept	24	VirtualWorld.BD_Plane Resource_4	Finished (#105)	False
VirtualWorld.Project_Plane Demand_1	Sent	25/24	agent: VirtualWorld.BD_Plane Resource_5	1	BeginMatching (#103) 1, 'Project_Plane Demand', 0.00p.
VirtualWorld.BD_Plane Resource_5	Accept	25/24	VirtualWorld.Project_Plane Demand_1	BeginMatching (#103)	1, 'Project_Plane Demand', 0.00p.
VirtualWorld.Project_Plane Demand_1	Sent	28	agent: VirtualWorld.BD_Plane Resource_5	1	Finished (#105) True
VirtualWorld.Project_Plane Demand_1	Sent	29/26	agent: VirtualWorld.BD_Plane Resource_6	1	BeginMatching (#103) 1, 'Project_Plane Demand', 0.00p.
VirtualWorld.BD_Plane Resource_5	Accept	28	VirtualWorld.Project_Plane Demand_1	Finished (#105)	True
VirtualWorld.BD_Plane Resource_6	Accept	29/26	VirtualWorld.Project_Plane Demand_1	BeginMatching (#103)	1, 'Project_Plane Demand', 0.00p.
VirtualWorld.Project_Plane Demand_1	Sent	32	agent: VirtualWorld.BD_Plane Resource_6	1	Finished (#105) True
VirtualWorld.Project_Plane Demand_1	Sent	33/30	agent: VirtualWorld.BD_Plane Resource_7	1	BeginMatching (#103) 1, 'Project_Plane Demand', 0.00p.
VirtualWorld.BD_Plane Resource_6	Accept	32	VirtualWorld.Project_Plane Demand_1	Finished (#105)	True
VirtualWorld.BD_Plane Resource_7	Accept	33/30	VirtualWorld.Project_Plane Demand_1	BeginMatching (#103)	1, 'Project_Plane Demand', 0.00p.
VirtualWorld.Project_Plane Demand_1	Sent	36	agent: VirtualWorld.BD_Plane Resource_7	1	Finished (#105) True
VirtualWorld.Project_Plane Demand_1	Sent	37/34	agent: VirtualWorld.BD_Plane Resource_8	1	BeginMatching (#103) 1, 'Project_Plane Demand', 0.00p.
VirtualWorld.BD_Plane Resource_7	Accept	36	VirtualWorld.Project_Plane Demand_1	Finished (#105)	True
VirtualWorld.BD_Plane Resource_8	Accept	37/34	VirtualWorld.Project_Plane Demand_1	BeginMatching (#103)	1, 'Project_Plane Demand', 0.00p.
VirtualWorld.Project_Plane Demand_1	Sent	40	agent: VirtualWorld.BD_Plane Resource_8	1	Finished (#105) True
VirtualWorld.Project_Plane Demand_1	Sent	41/38	agent: VirtualWorld.BD_Plane Resource_9	1	BeginMatching (#103) 1, 'Project_Plane Demand', 0.00p.
VirtualWorld.BD_Plane Resource_8	Accept	40	VirtualWorld.Project_Plane Demand_1	Finished (#105)	True
VirtualWorld.BD_Plane Resource_9	Accept	41/38	VirtualWorld.Project_Plane Demand_1	BeginMatching (#103)	1, 'Project_Plane Demand', 0.00p.
VirtualWorld.BD_Plane Resource_9	Sent	43	agent: VirtualWorld.Project_Plane Demand_1	5	Finished (#105) False
VirtualWorld.Project_Plane Demand_1	Accept	43	VirtualWorld.BD_Plane Resource_9	Finished (#105)	False
VirtualWorld.Project_Plane Demand_1	Sent	45/43	agent: VirtualWorld.BD_Plane Resource_5	1	Reserve (#106)
VirtualWorld.BD_Plane Resource_5	Accept	45/43	VirtualWorld.Project_Plane Demand_1	Reserve (#106)	

Рис. 56. Окно лога сообщений в процессе матчинга заказ/ресурс

- ◆ Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*).
- ◆ Выполните очистку лога (вызов контекстного меню *-> Clear log*).
- ◆ Отмените фильтры *GetAgent* и *SendAgent*.
- ◆ При помощи кнопки  вновь запустите сцену на моделирование.
- ◆ Наблюдайте в окне системного лога обмен сообщениями между агентами в процессе матчинга (количество выдаваемых сообщений существенно уменьшится).
- ◆ Если далее не предполагается использование системного лога, закройте или сверните его окно.

2.5.5 Изменение значений атрибутов агентов в сцене

- ◆
- ◆ Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*).
- ◆ У агента проекта *Project_Plane_Demand_1* измените допустимую границу поиска прототипа по дальности полета на 10%, т.е. установите значения атрибута *Percent of the Deviation Distance* равной 0.1
- ◆ 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. 57, а таблица принятия решения агента проекта – на рис. 58. Ограничениям по количеству пассажиров и ограничениям на дальность полета (1800 км – 2000 км) теперь соответствуют только самолеты прототипы 3, 7 и 8. По критерию максимальной дальности полета агент проекта зарезервирует прототип 3 – ВАЕ 146-200.

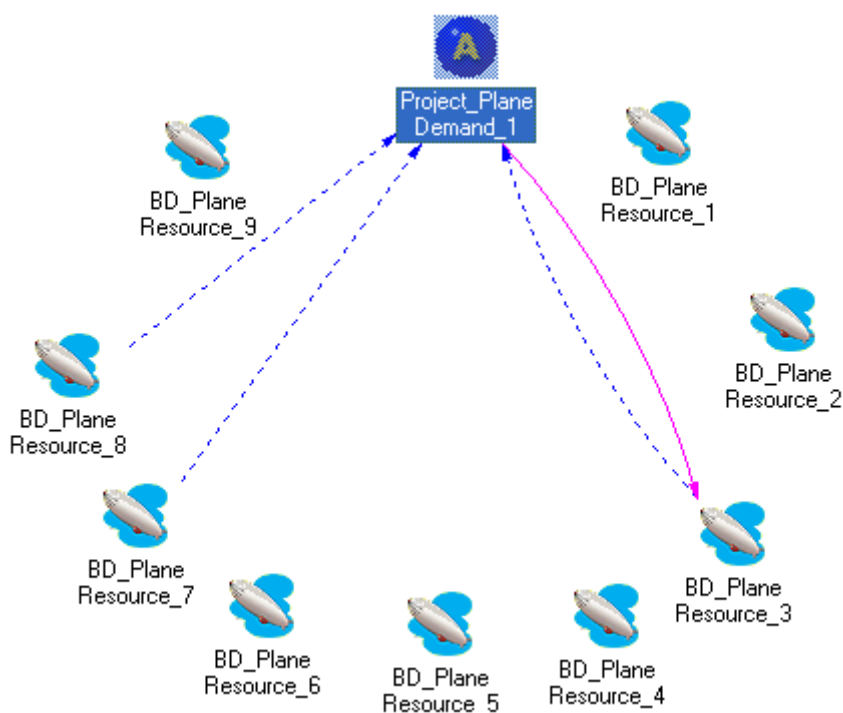


Рис. 57. Результаты матчинга при изменении значения атрибута агента

Agent	BD_Distance
BD_Plane Resource_3	2180
BD_Plane Resource_7	2020
BD_Plane Resource_8	2000

Рис. 58. Таблица принятия решений агента *Project_Plane Demand_1*

В результате матчинга выполнена следующая операция резервирования:

- *Project_Plane Demand_1 – BD_Plane Resource_3.*


◆

- ◆ Сохраните сцену под именем *Scene_TTT_Distance ontology_2.osf*.
- ◆ Завершите работу с исполняющей системой (*File -> Close*).


◆

2.5.6 Добавление агентов в сцену

◆

- ◆ Загрузите сцену *Scene_TTT_Distance ontology_1.osf* (кнопка  или последовательность команд *Tools - > Load ontology scene...*)
- ◆ Добавьте в сцену агента самолета-прототипа со следующими значениями атрибутов

<i>Имя агента</i>	<i>Тип самолета</i>	<i>Дальность полета, км</i>	<i>Количество пассажиров</i>	<i>Взлетная масса, кг</i>
BD_Plane Resource_10	RJ 115	2295	110	46000

- ◆ 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. Атрибуты агента, введенного в сцену, соответствуют ограничениям. Но по критерию максимальной дальности полета резервируется по-прежнему агент самолета прототипа 5 - Fokker 100 (рис. 59).

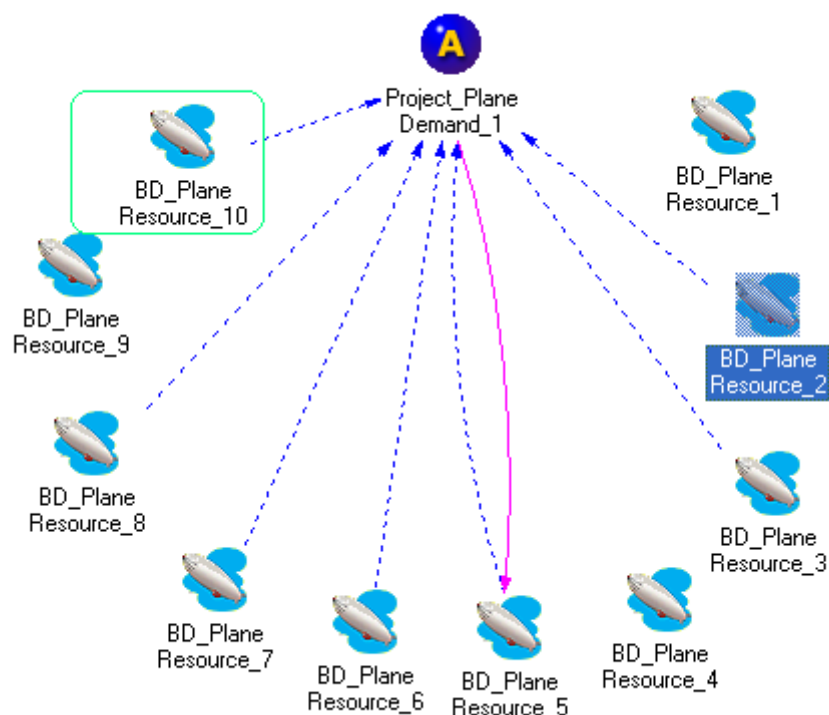


Рис. 59. Результаты матчинга при добавлении агента в сцену

В результате матчинга выполнена следующая операция резервирования:

- *Project_Plane Demand_1 – BD_Plane Resource_5.*

◆

◆ Сохраните сцену под именем *Scene_TTT_Distance ontology_3.osf*.

◆ Завершите работу с исполняющей системой (*File -> Close*).

◆

3. ВЫБОР ТИПА САМОЛЕТА ПО КРИТЕРИЮ ВЗЛЕТНОЙ МАССЫ

3.1. Постановка задачи

На основании заданных тактико-технических требований для проектируемого самолета необходимо подобрать в базе данных самолет, отвечающий заданным требованиям. Назначение самолета – пассажирский, либо грузовой.

ТТТ для пассажирского самолета:

- дальность полета, км;
- число пассажиров, чел.;
- взлетная масса, кг.

ТТТ для грузового самолета:

- дальность полета, км;

- масса целевой нагрузки, кг;
- взлетная масса, кг.

Заданы также ограничения снизу на дальность полета и количество пассажиров.

Необходимо выбрать в базе данных прототипы, отвечающие ТТТ по дальности полета и количеству пассажиров. Среди этих прототипов необходимо выбрать самолет, обеспечивающий минимальную взлетную массу. Будем рассматривать прототипы, выбранные при решении задачи из раздела 2.

Ниже приведен пример выбора пассажирского самолета. Выбор грузового самолета производится аналогично.

3.2 Проектирование дескриптивной онтологии

3.2.1 Создание онтологии

Для создания дескриптивной онтологии воспользуемся ранее созданной онтологией предметной области “Тактико-технические требования к проектируемому самолету”. В этой онтологии теперь не надо учитывать процент отклонений по дальности полета и количеству пассажиров и, соответственно рассчитывать минимальные и максимальные значения этих атрибутов. В данной задаче будут задаваться только нижние границы значений дальности полета и количества пассажиров. В онтологию следует добавить атрибут “Взлетная масса самолета-прототипа”.

◆

- ◆ Загрузите конструктор онтологий (файл *OntCons.exe*).
- ◆ Откройте ранее созданную онтологию предметной области “Тактико-технические требования к проектируемому самолету” (*File -> Open -> TTT_Distance ontology*).
- ◆ Сохраните библиотеку онтологий *TTT_Distance ontology* под именем *TTT_Mass ontology* (*File -> Save as -> TTT_Mass ontology*). Далее работайте с этой библиотекой онтологий.
- ◆ В дереве концептов выделите дескриптивную онтологию и переименуйте ее, вводя в поле *Name* имя *Ontology_TTT_Mass* (рис. 60).

◆

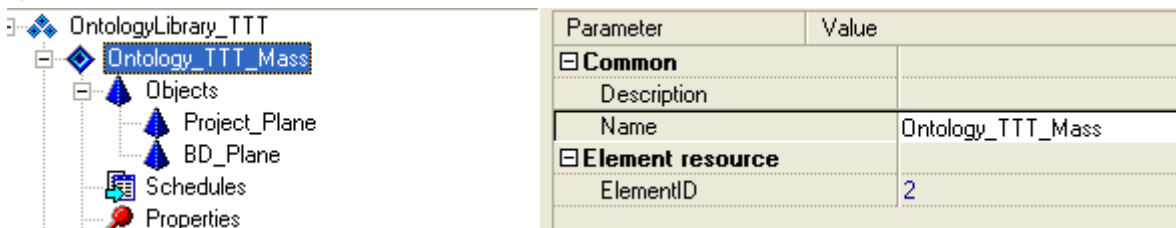


Рис. 60. Переименование дескриптивной онтологии

3.2.2 Редактирование дерева концептов онтологии

- ◆
- ◆ Удалите из дерева концептов следующие атрибуты концепта “Проектируемый самолет” (рис. 61): *Percent of the Deviation_Distance*, *Project_Min Distance*, *Project_Max Distance*, *Percent of the Deviation_Number of Passengers*, *Project_Min Number of Passengers*, *Project_Max Number of Passengers*.

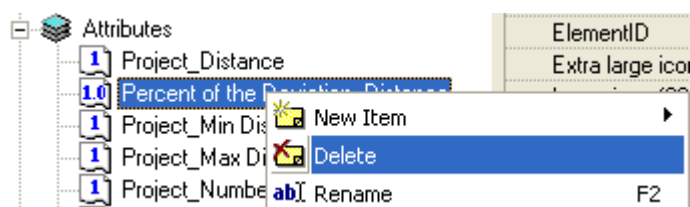


Рис. 61 – Удаление атрибутов из дерева концептов онтологии

- ◆ Удалите из дерева концептов все скрипты, т.к. в данной постановке задачи не требуется рассчитывать нижнюю границу значений допустимой дальности полета и допустимого количества пассажиров, а верхние границы этих атрибутов вообще не задаются.
- ◆ Создайте целый атрибут *BD_Mass* (*Attributes* -> *New Item* -> *Integer Attribute*), переименуйте созданный атрибут в *BD_Mass*. Свяжите этот атрибут с концептом *BD_Plane*.

◆
Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Тактико-технические требования к проектируемому самолету” приведено на рис. 62.

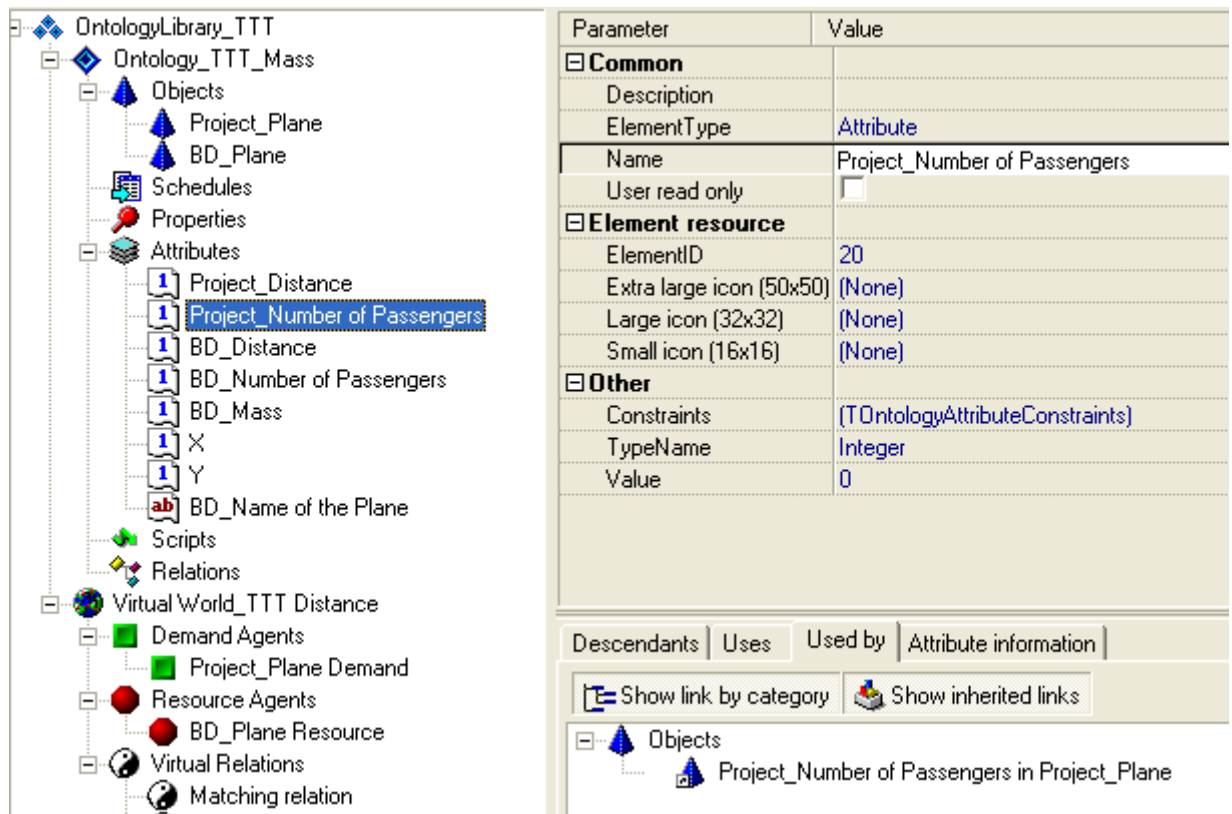


Рис. 62. Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “ТТТ к проектируемому самолету”

◆

3.3 Проектирование онтологии мира заказов и ресурсов

3.3.1 Редактирование условий матчинга

◆

◆ В дереве концептов онтологии мира заказов и ресурсов перейдите в закладку *Used by* (*Virtual relations* -> *Matching relation* -> *Used by*) и выделите *Matching relation* (*Project_Plane Demand*, *BD_Plane Resource*), далее в контекстном меню выберите *Edit virtual relation properties*. При этом откроется окно редактирования условий матчинга.

◆ Удалите условия матчинга (ограничения), помеченные знаком ✕ (рис. 63).

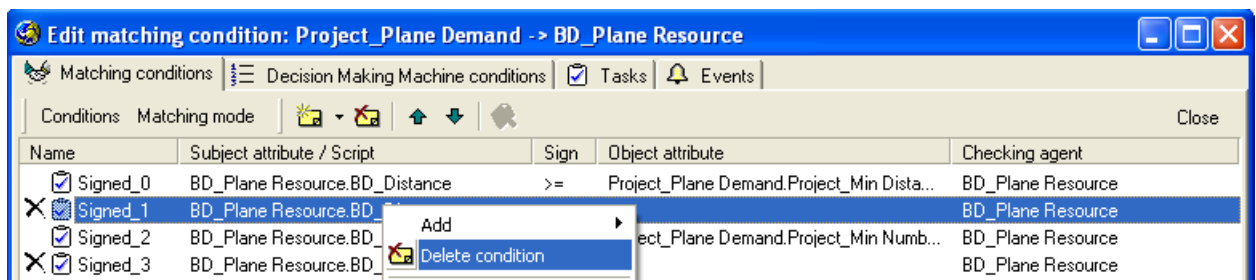


Рис. 63. Удаление условий матчинга

◆ Ограничения данной задачи показаны на рис. 64.

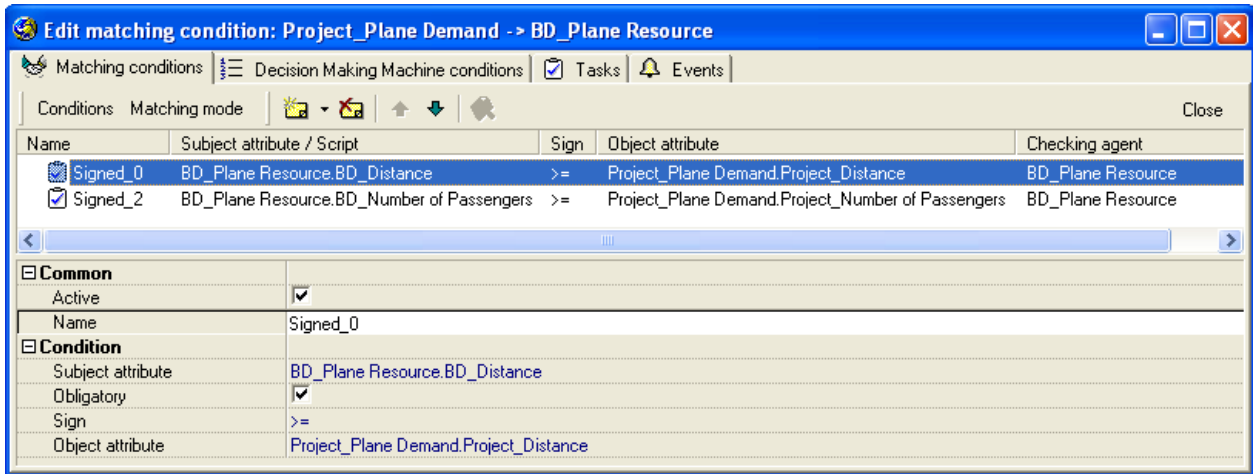


Рис. 64. Условия матчинга *Project_Plane Demand -> BD_Plane Resource*

◆

3.3.2 Создание условия принятия решения - минимизация взлетной массы самолета-прототипа

◆

- ◆ Перейдите в закладку *Decision Making Machine Conditions*.
- ◆ Удалите условие максимизации дальности полета (рис. 65).

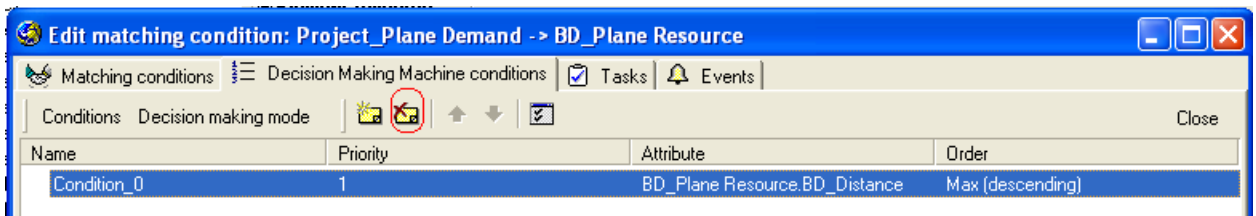



Рис. 65. Удаление условия принятия решения

- ◆ Создайте условие принятия решений для матчинга *Project_Plane Demand -> BD_Plane Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия:

- Attribute = 'BD_Plane Resource.BD_Mass';
- Order = 'Min';
- Weight = '100'.

Условие означает, что агент проектируемого самолета при матчинге более, чем с одним агентом самолета-прототипа, будет выбирать самолет-прототип с минимальной взлетной массой.

- ◆ Активизируйте условие принятия решения, выставив флажок *Active*.

◆

Условие принятия решения в матчинге для поиска прототипа по

взлетной массе приведено на рис. 66.

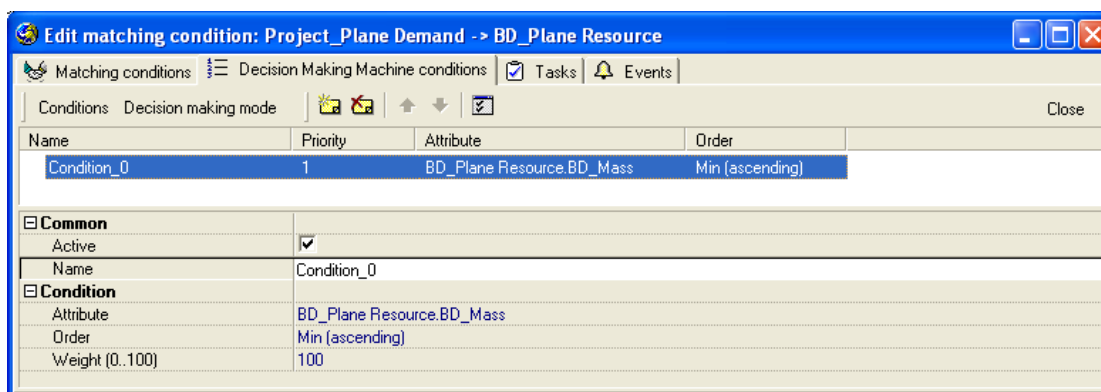



Рис. 66. Условие принятия решения в матчинге *Project_Plane Demand – BD_Plane Resource*

◆

3.3.3 Сохранение онтологий предметной области “ТТТ к проектируемому самолету”

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

◆

3.4 СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ СЦЕНЫ

◆

- ◆ Создайте новую онтологическую сцену (*File → New scene -> Load ontology*, выберите онтологию *TTT_Mass ontology.ocl*).


◆

- ◆ В окне физического мира создайте одного агента проектируемого самолета и девять агентов самолетов-прототипов в базе данных. Минимально допустимая дальность полета 2000 км, минимально допустимое количество пассажиров 100 чел. С помощью инспектора агентов установите следующие значения атрибутов для агентов:

<i>Имя агента</i>	<i>Тип самолета</i>	<i>Дальность полета, км</i>	<i>Количество пассажиров</i>	<i>Взлетная масса, кг</i>
Project_Plane Demand_1		2000	100	
BD_Plane Resource_1	Бритиш Аэро-спейс 146 - 300	2000	122	44230

BD_Plane Resource_2	Бритиш Аэро-спейс 146 - 100	1730	93	38010
BD_Plane Resource_3	Бритиш Аэро-спейс 146 - 200	2180	109	42180
BD_Plane Resource_4	Фоккер 70	2000	79	36740
BD_Plane Resource_5	Фоккер 100	2390	107	43090
BD_Plane Resource_6	Аеро Интернешнл Ридженд RJ100	2260	100	46000
BD_Plane Resource_7	ТУ - 134Б	2020	96	47600
BD_Plane Resource_8	ТУ – 334 - 100	2000	110	46100
BD_Plane Resource_9	ЯК – 42Д	2150	120	56500

3.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА

- ◆
- ◆ 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.
- ◆

Процесс матчинга начинается с того, что агент проекта (*Project_Plane Demand_1*) проверяет значения атрибутов агентов самолетов-прототипов и выбирает агентов, у которых значения атрибутов соответствуют ограничениям на минимально и максимально допустимые значения дальности полета и количества пассажиров. Дальность полета должна быть не меньше 2000 км, а количество пассажиров – не меньше 100 человек.

Прототип 2 не соответствует ограничениям по дальности полета и количеству пассажиров, прототипы 4 и 7 не соответствуют ограничению по количеству пассажиров. Поэтому агент проектируемого самолета не соглашается на матчинг с ними (синие пунктирные стрелки в результатах матчинга отсутствуют). Далее агент проекта строит таблицу принятия решений, в которой размещает прототипы по возрастанию взлетной массы, и выбирает из таблицы лучший самолет-прототип *BD_Plane_3* по критерию минимальной взлетной массы. Агент проекта резервирует агента прототипа *BD_Plane_3*, который удовлетворяет его требованиям: *Project_Plane*

Demand_1 – *BD_Plane_3* (сплошная малиновая линия в результатах матчинга).

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 67), в которой перечислены все агенты самолетов-прототипов, с которыми возможен матчинг в соответствии с критерием принятия решения (минимальной взлетной массой).

Agent	BD_Mass
BD_Plane Resource_3	42180
BD_Plane Resource_5	43090
BD_Plane Resource_1	44230
BD_Plane Resource_6	46000
BD_Plane Resource_8	46100
BD_Plane Resource_9	56500

Рис. 67. Таблица принятия решений агента *Project_Plane Demand_1*

Аналогично можно посмотреть структуру агента-ресурса (рис. 68). В правой части таблицы указаны атрибуты агента-ресурса (*Simple*), а также атрибуты агента-проекта, которые использовались в процессе принятия решения о резервировании (*Partner*).

Partner	Status	Name	Value	State	Type
Project_Plane Demand_1	Passive	BD_Number of Passengers	109	Assigned	Simple
		Project_Number of Passengers	100	Assigned	Partner
		BD_Distance	2180	Assigned	Simple
		Project_Distance	2000	Assigned	Partner
		BD_Mass	42180	Assigned	Simple

Рис. 68. Структура агента ресурса *BD_Plane Resource_3*

Окончательные результаты матчинга можно видеть на рисунке 69. В результате матчинга была выполнена следующая операция резервирования:

- *Project_Plane Demand_1* – *BD_Plane Resource_3*.

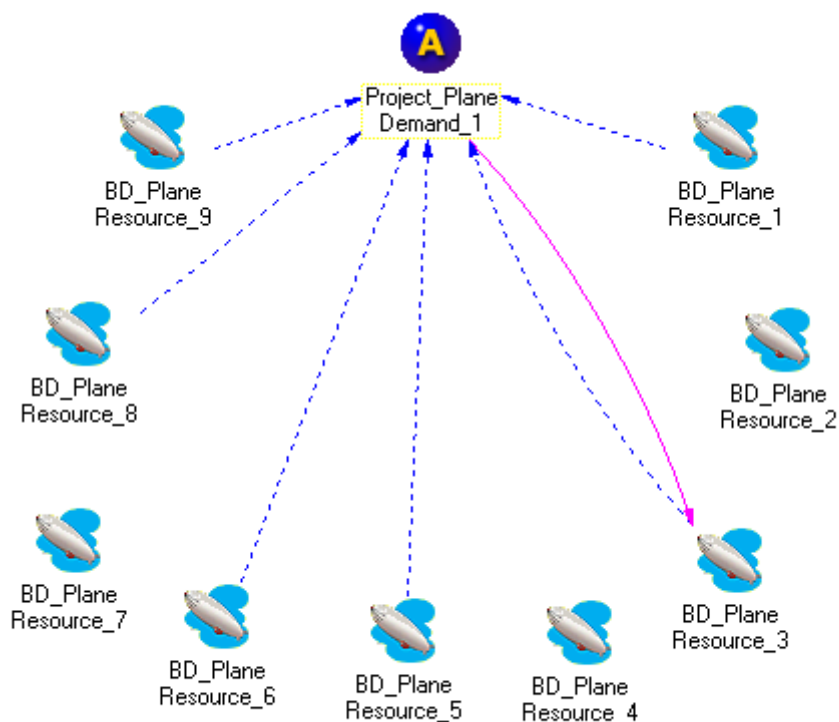



Рис. 69. Результаты матчинга

- ◆
- ◆ С помощью кнопки  или последовательности команд *Tools - > Save ontology scene...* сохраните сцену под именем *Scene_ TTT_Mass_ontology.osf*.
- ◆ Завершите работу с исполняющей системой (*File -> Close*).
- ◆

4 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 создать концепт онтологии мира заказов/ресурсов и задать его

свойства (на примере концептов «заказ» и «ресурс»)?

7. Как установить отношение матчинга между концептами мира заказов/ресурсов?

8. В чем отличие двустороннего матчинга от одностороннего? Может ли ресурс инициировать матчинг?

9. Как задать условие матчинга между концептами мира заказов/ресурсов? Приведите пример знакового условия матчинга.

10. Как сконструировать онтологическую сцену с помощью инструментов, предоставляемых исполняющей системой?

11. Как изменить значения атрибутов агентов с помощью Инспектора агентов? Как добавить агентов в готовую онтологическую сцену?

12. Какие действия необходимо выполнить в процессе моделирования онтологической сцены (на примере сцены из раздела 2.5.1)?

13. Проанализируйте, как выполняется принятие решений при моделировании онтологической сцены из раздела 2.5.1.

14. Для чего предназначен системный лог и логи агентов? Какую информацию можно получить с их помощью (на примере онтологической сцены из раздела 2.5.4)?

5 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ