

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

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

MAGENTA CORPORATION LIMITED

***Н.М.Боргест, Е.В. Симонова***

**ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ ПРИ ВЫБОРЕ  
УДЕЛЬНОЙ НАГРУЗКИ НА КРЫЛО**

Методические указания к лабораторной работе № 2 по курсу

**“ОНТОЛОГИЯ ПРОИЗВОДСТВЕННОЙ СФЕРЫ”**

Самара  
2008

<b>1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ.....</b>	<b>4</b>
<b>2 ВЫБОР УДЕЛЬНОЙ НАГРУЗКИ НА КРЫЛО.....</b>	<b>5</b>
2.1. ПОСТАНОВКА ЗАДАЧИ.....	5
2.1.1 Расчет условия обеспечения заданной скорости захода на посадку.....	5
2.1.2 Расчет условия обеспечения заданной крейсерской скорости на расчетной высоте полета.....	6
2.1.3 Расчет удельной нагрузки на крыло.....	7
2.2 ПРОЕКТИРОВАНИЕ ДЕСКРИПТИВНОЙ ОНТОЛОГИИ.....	8
2.2.1 Создание онтологии.....	8
2.2.2 Создание и удаление концепта.....	8
2.2.2.2 Концепт «атрибут».....	9
2.2.2.3 Создание связей между концептами.....	10
2.2.3 Концепт «скрипт».....	12
2.2.3.1 Определение скрипта для вычисления крейсерской скорости полета.....	13
2.2.3.2 Определение скрипта для вычисления относительной массы топлива.....	14
2.2.3.3 Определение скрипта для вычисления удельной нагрузки на крыло при заходе на посадку.....	15
2.2.3.4 Определение скрипта для вычисления относительной плотности воздуха на расчетной высоте.....	16
2.2.3.5 Определение скрипта для вычисления удлинения фюзеляжа.....	17
2.2.3.6 Определение скрипта для вычисления коэффициента лобового сопротивления при нулевой подъемной силе.....	18
2.2.3.7 Определение скрипта для вычисления удельной нагрузки на крыло при заданной крейсерской скорости на расчетной высоте полета.....	19
2.2.3.8 Определение скрипта для вычисления удельной нагрузки на крыло.....	20
2.2.3.9 Определение скрипта для вычисления максимального аэродинамического качества.....	22
2.2.4 Представление онтологии в виде семантической сети.....	23
2.3 ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ МИРА ЗАКАЗОВ И РЕСУРСОВ.....	24
2.3.1 Создание онтологии мира заказов и ресурсов.....	24
2.3.2 Создание концепта «агент заказа».....	26
2.3.3 Создание концепта «агент ресурса».....	26
2.3.4 Виртуальные отношения: отношение матчинга.....	27
2.3.4 Условия матчинга.....	28
2.3.5 Условия принятия решения (Decision Making Machine conditions).....	29
2.3.5.1 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.....	29
2.3.5.2 Создание условия принятия решения - максимизация аэродинамического качества.....	30
2.3.6 Параметры представления экземпляра объекта в сцене.....	31
2.3.7 Сохранение онтологий предметной области “Выбор удельной нагрузки на крыло”.....	32
2.4 СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ СЦЕНЫ.....	33

2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА .....	34
2.5.1 Запуск сцены на моделирование для самолета ТУ-154.....	34
2.5.2 Сохранение сцены виртуального мира .....	36
2.5.3. Загрузка ранее созданной сцены виртуального мира.....	36
2.5.4 Моделирование сцены для самолета ИЛ-86.....	37
2.5.5 Моделирование сцены для самолета ЯК-40.....	38
<b>4 КОНТРОЛЬНЫЕ ВОПРОСЫ .....</b>	<b>39</b>
<b>5 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....</b>	<b>39</b>

# 1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

На основании нижеприведенных характеристик самолетов-прототипов, информация о которых хранится в базе данных, необходимо рассчитать удельную нагрузку на крыло.

Характеристики самолетов-прототипов:

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

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 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.

Цель лабораторной работы № 2 – разработка мультиагентного приложения, реализующего возможности скриптов 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. Идентификаторы концептов, атрибутов и скриптов записываются английским шрифтом. Действия, которые непосредственно предлагается выполнить пользователю, выделяются знаками, показанными ниже. Будьте, пожалуйста, внимательны!



## 2 ВЫБОР УДЕЛЬНОЙ НАГРУЗКИ НА КРЫЛО

### 2.1. Постановка задачи

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

Условие обеспечения заданной скорости захода на посадку рассчитывается по формуле (1)

$$p'_0 \leq \frac{C_{y_{\max \text{noc}}} \cdot V_{3.П.}^2}{30,2 \cdot (1 - \overline{m_T})}, \quad (1)$$

где

- $p'_0$  - удельная нагрузка на крыло при заходе на посадку,  $\text{даН/м}^2$ ;
- $C_{y_{\max \text{noc}}}$  - значение максимального коэффициента подъемной силы самолета в посадочной конфигурации; значение  $C_{y_{\max \text{noc}}}$  выбирается на основе статистических данных в зависимости от системы механизации крыла. Для упрощения решения задачи принято:
  - для эффективной механизации  $C_{y_{\max \text{noc}}} = 2.5$ ;
  - для слабой (неэффективной) механизации  $C_{y_{\max \text{noc}}} = 2.0$ .
- $V_{3.П.}$  - скорость захода на посадку, м/с; значение  $V_{3.П.}$  выбирается на основе статистических данных. В данной работе принято значение:  $V_{3.П.} = 230 \text{ км/ч} = 63,89 \text{ м/с}$ ;
- $\overline{m_T}$  - 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) и (3) в зависимости от выбранной эффективной механизации (исходное данное)

- для эффективной механизации  $p'_0 = 10204.83 / (30.2(1 - \overline{m_T}))$ ; (2)

- для неэффективной механизации  $p'_0 = 8163.86 / (30.2(1 - \overline{m_T}))$  (3)

Предполагаемое значение относительной массы топлива рассчитывается по формуле (4)

$$\overline{m_T} = 0.075 + \frac{0.0175 \cdot L_p}{V_{кр}}, \quad (4)$$

где

- $L_p$  – расчетная дальность полета, км (исходное данное);

- $V_{кр}$  - крейсерская скорость полёта, м/с.

Крейсерская скорость полета рассчитывается по формуле (5)

$$V_{кр} = M * 20.04679 * \sqrt{288.15 - 41318.979 * H_{кр} / (6356766 + H_{кр})}, \quad (5)$$

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 (6)

$$P_0'' \leq \frac{0.208 * \Delta H_{кр} * V_{кр}^2 * \sqrt{C_{x0}}}{1 - 0.6 * m_t}, \quad (6)$$

где

- $P_0''$  - удельная нагрузка на крыло при заданной крейсерской скорости на расчетной высоте полета,  $даН/м^2$  ;
- $V_{кр}$  - крейсерская скорость полёта, м/с, рассчитывается по формуле (5);
- $m_t$  - предполагаемое значение относительной массы топлива, рассчитывается по формуле (4);
- $\Delta H_{кр}$  - относительная плотность воздуха на расчетной высоте, м;
- $C_{x0}$  - коэффициент лобового сопротивления при нулевой подъемной силе.

Относительная плотность воздуха на расчетной высоте определяется по формуле (7)

$$\Delta H_{кр} = 0.8 - 0.000046 * H_{кр}, \quad (7)$$

где

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

Коэффициент лобового сопротивления при нулевой подъемной силе рассчитывается по формуле (8)

$$C_{x0} = (0,882 + 0,147M) \cdot \left[ 0,015288 + 0,000656 \cdot \lambda_{\phi} + \frac{0,041}{\lambda_{\phi}^2} \right]. \quad (8)$$

где

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

Удлинение фюзеляжа рассчитывается по формуле (9)

$$\lambda_{\phi} = \frac{0.906 * n_{vr} + 1.323 * n_{pas}}{(0.649 * n_{vr} + 0.6) * n_{vr}}. \quad (9)$$

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 (10)

$$K_{max} = \frac{1}{0,43\sqrt{C_{x0}}} + 1. \quad (10)$$

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 (11)

$$P_0 = \min (p'_0, p''_0). \quad (11)$$

где

- $P_0$  – расчетное значение удельной нагрузки на крыло,  $даН/м^2$ ;
- $p'_0$  - удельная нагрузка на крыло при заходе на посадку,  $даН/м^2$ ; рассчитывается по формуле (1);

- $p_0''$  - удельная нагрузка на крыло при заданной крейсерской скорости на расчетной высоте полета,  $даН/м^2$ ; рассчитывается по формуле (6).

## 2.2 Проектирование дескриптивной онтологии

### 2.2.1 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*.
- ◆

### 2.2.2 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*. Созданный концепт можно удалить, выделив его и нажав клавишу <Del>, 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 «Выбор удельной нагрузки на крыло» необходимо создать два концепта «объект»:

- *Wing\_Demand* (проектируемый самолет) с атрибутами
  - *\_Effective\_mehan* (эффективная механизация крыла: да/нет);
  - *\_H\_kr* (высота полета);
  - *\_M* (число Маха);
  - *\_L* (расчетная дальность полета);
  - *\_n\_pas* (количество пассажиров);



- $n_{vr}$  (количество пассажиров в ряду);
- $Motn_{topl}$  (относительная масса топлива);
- $V_{kr}$  (крейсерская скорость полета);
- $\Delta H_{kr}$  (относительная плотность воздуха на расчетной высоте);
- $CX_0$  (коэффициент лобового сопротивления при нулевой подъемной силе);
- $\lambda_{fuz}$  (удлинение фюзеляжа);
- $PO_{zahpos}$  (удельная нагрузка на крыло при заходе на посадку);
- $PO_H$  (удельная нагрузка на крыло при заданной крейсерской скорости на расчетной высоте полета);
- $PO$  (удельная нагрузка на крыло);
- $K_{max}$  (максимальное аэродинамическое качество).

Примечание: имена атрибутов, которые являются исходными данными для расчета, начинаются с нижнего подчеркивания.

- *Wing\_Resource* - объект-ресурс в данной задаче не имеет атрибутов, он необходим только для того, чтобы запустить матчнинг, в процессе которого будут выполнены необходимые расчеты.

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



- ◆ 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 .
- ◆ Создайте объект *Wing\_Resource* (аналогично созданию *Wing\_Demand*), переименуйте его в *Wing\_Resource*, 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).



- ◆ Создайте булевский атрибут *\_Effective\_mehan* (*Attributes -> New Item -> Boolean Attribute*), переименуйте созданный атрибут в *\_Effective\_mehan*.
- ◆ Создайте целый атрибут *\_H\_kr* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_H\_kr*.
- ◆ Создайте вещественный атрибут *\_M* (*Attributes ->New Item -> Float Attribute*), переименуйте созданный атрибут в *\_M*.
- ◆ Создайте целый атрибут *\_L* (*Attributes ->New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_L*.
- ◆ Создайте целый атрибут *\_n\_pas* (*Attributes ->New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_n\_pas*.
- ◆ Создайте целый атрибут *\_n\_vr* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_n\_vr*.
- ◆ Создайте вещественный атрибут *Motn\_topl* (*Attributes ->New Item -> Float Attribute*), переименуйте созданный атрибут в *Motn\_topl*.
- ◆ Создайте вещественный атрибут *V\_kr* (*Attributes ->New Item -> Float Attribute*), переименуйте созданный атрибут в *V\_kr*.
- ◆ Создайте вещественный атрибут *DeltaH\_kr* (*Attributes ->New Item -> Float Attribute*), переименуйте созданный атрибут в *DeltaH\_kr*.
- ◆ Создайте вещественный атрибут *K\_max* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *K\_max*.
- ◆ Создайте вещественный атрибут *CX\_0* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *CX\_0*.
- ◆ Создайте вещественный атрибут *Lambda\_fuz* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *Lambda\_fuz*.
- ◆ Создайте вещественный атрибут *P0\_zahpos* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *P0\_zahpos*.
- ◆ Создайте вещественный атрибут *P0\_H* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *P0\_H*.
- ◆ Создайте вещественный атрибут *P0* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *P0*.
- ◆ Создайте целый атрибут *X* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *X*.
- ◆ Создайте целый атрибут *Y* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *Y*.
- ◆

### 2.2.2.3 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*.

- ◆
- ◆ Установите связи между атрибутами *\_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* и объектом *Wing\_Demand*.
- ◆ Установите связи между атрибутами *X*, *Y* и объектом *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 рис. 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.

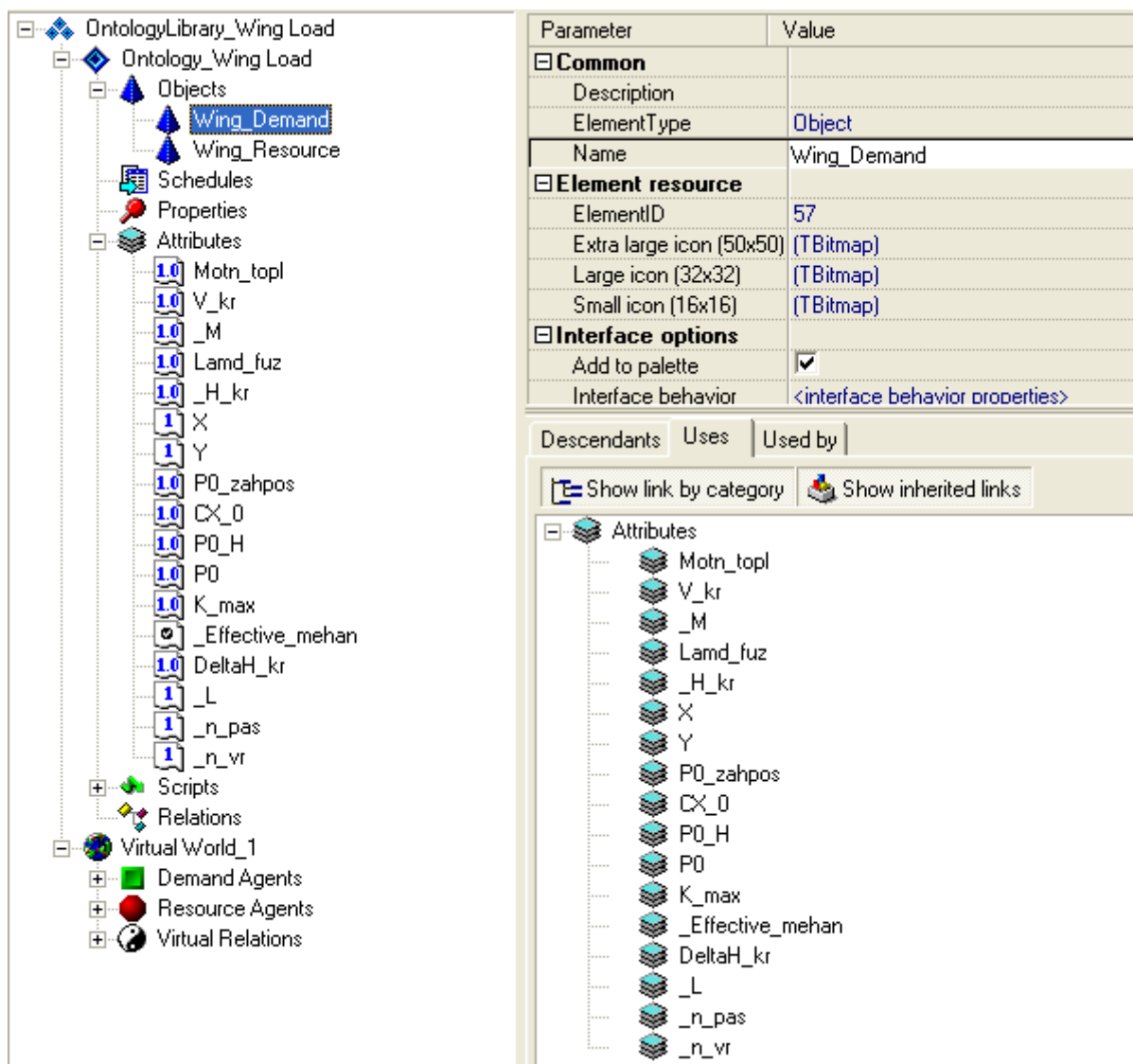



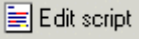
Рис. 1. Атрибуты (свойства) объекта *Wing\_Demand* в закладке *Uses*

### 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. 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*).

### 2.2.3.1 Определение скрипта для вычисления крейсерской скорости полета


Чтобы вычислить с помощью скрипта крейсерскую скорость полета по формуле (5), необходимо выполнить следующие действия:

- ◆
- ◆ Создайте 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).
- ◆ Напишите тело скрипта (т.е., собственно скрипт). 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.
- ◆

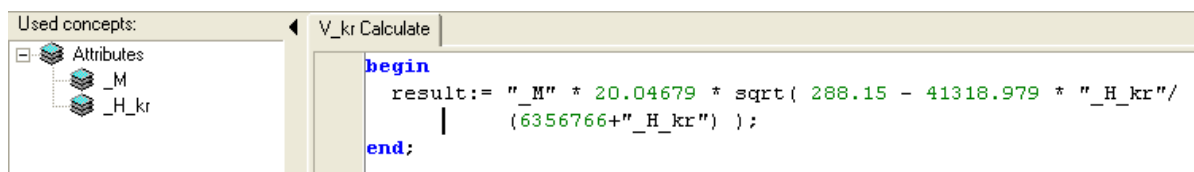


Рис. 2. Атрибуты и тело скрипта *V\_kr Calculate*

◆

### 2.2.3.2 Определение скрипта для вычисления относительной массы топлива

Чтобы вычислить с помощью скрипта относительную массу топлива по формуле (4), необходимо выполнить следующие действия:


◆

- ◆ Создайте концепт «скрипт», который будет вычислять относительную массу топлива. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.


- ◆ Созданный скрипт переименуйте в *Motn\_topl Calculate* и свяжите его с атрибутом *Motn\_topl*, т.е. перетащите скрипт на атрибут *Motn\_topl*.

- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *Motn\_topl Calculate*, вычисляющего относительную массу топлива, являются атрибуты *\_L* и *V\_kr* (рис. 3).

◆

- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Motn\_topl 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.

```
begin
    Result := 0.075+(0.0175*"_L")/"V_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.

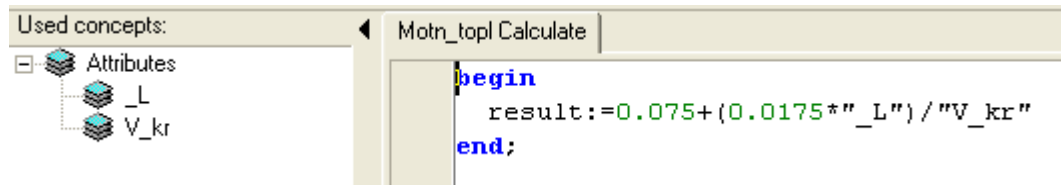



Рис. 3. Атрибуты и тело скрипта *Motn\_topl Calculate*

◆

### 2.2.3.3 Определение скрипта для вычисления удельной нагрузки на крыло при заходе на посадку

Чтобы вычислить с помощью скрипта удельную нагрузку на крыло при заходе на посадку по формулам (2) и (3), необходимо выполнить следующие действия:

◆


- ◆ Создайте концепт «скрипт», который будет вычислять удельную нагрузку на крыло при заходе на посадку. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *P0\_zahpos Calculate* и свяжите его с атрибутом *P0\_zahpos*, т.е. перетащите скрипт на атрибут *P0\_zahpos*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *P0\_zahpos Calculate*, вычисляющего удельную нагрузку на крыло при заходе на посадку, являются атрибуты *\_Effective\_mehan* и *Motn\_topl* (рис. 4).
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *P0\_zahpos Calculate* в дереве концептов дескриптивной онтологии, затем перейдите в закладку *Script body*, нажмите на кнопку  *Edit script*. При этом откроется окно редактора скриптов, в котором необходимо набрать следующий текст.

```
begin
  if "_Effective_mehan"
```

```

then result := 10204/(30.2 * (1-"Motn_top1"))
else result := 8163/(30.2 * (1-"Motn_top1"));
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.

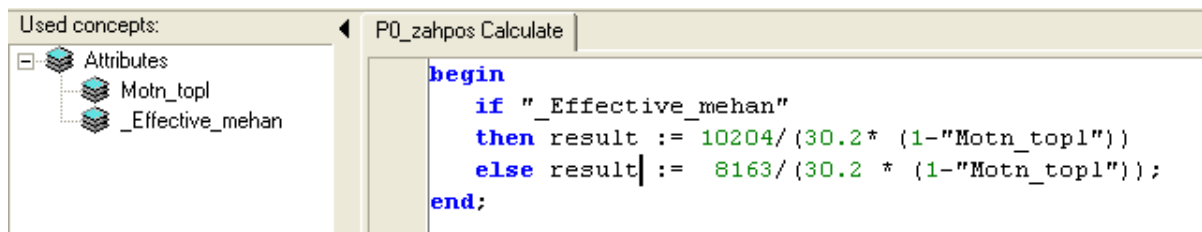



Рис. 4. Атрибуты и тело скрипта *P0\_zahpos Calculate*

◆

#### 2.2.3.4 Определение скрипта для вычисления относительной плотности воздуха на расчетной высоте

Чтобы вычислить с помощью скрипта относительную плотность воздуха на расчетной высоте по формуле (7), необходимо выполнить следующие действия:


◆

- ◆ Создайте концепт «скрипт», который будет вычислять относительную плотность воздуха на расчетной высоте. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *DeltaH\_kr Calculate* и свяжите его с атрибутом *DeltaH\_kr*, т.е. перетащите скрипт на атрибут *DeltaH\_kr*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *DeltaH\_kr Calculate*, вычисляющего относительную плотность воздуха на расчетной высоте, является атрибут *\_H\_kr* (рис. 5).
- ◆
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Motn\_top1 Calculate* в дереве концептов дескриптивной онтологии, затем перейдите в закладку *Script body*, нажмите на кнопку  *Edit script*. При



этом откроется окно редактора скриптов, в котором необходимо набрать следующий текст.

```
begin  
    Result := 0.8 - 0.000046 * "_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.

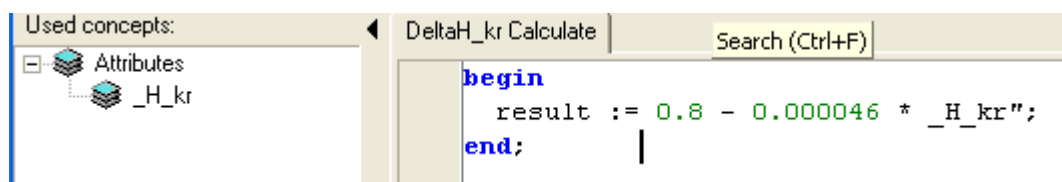


Рис. 5. Атрибуты и тело скрипта *DeltaH\_kr Calculate*

◆


#### 2.2.3.5 Определение скрипта для вычисления удлинения фюзеляжа

Чтобы вычислить с помощью скрипта удлинение фюзеляжа по формуле (9), необходимо выполнить следующие действия:


◆

- ◆ Создайте концепт «скрипт», который будет вычислять удлинение фюзеляжа. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *Lambda\_fuz Calculate* и свяжите его с атрибутом *Lambda\_fuz*, т.е. перетащите скрипт на атрибут *Lambda\_fuz*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *Lambda\_fuz Calculate*, вычисляющего удлинение фюзеляжа, являются атрибуты *\_n\_pas* и *\_n\_vr* (рис. 6).

◆

- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Lambda\_fuz 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.

```
begin
    Result := ( 0.906 * "_n_vr" + 1.323 * "_n_pas" ) /
              ( ( 0.649 * "_n_vr" + 0.6 ) * "_n_vr" );
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.

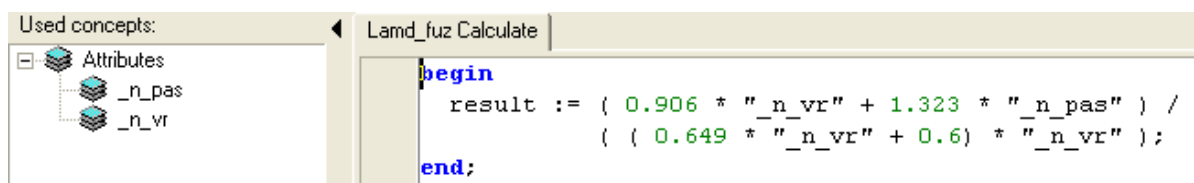


Рис. 6. Атрибуты и тело скрипта *Lambda\_fuz Calculate*

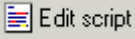
◆

### 2.2.3.6 Определение скрипта для вычисления коэффициента лобового сопротивления при нулевой подъемной силе

Чтобы вычислить с помощью скрипта коэффициент лобового сопротивления при нулевой подъемной силе по формуле (8), необходимо выполнить следующие действия:

◆


- ◆ Создайте концепт «скрипт», который будет вычислять коэффициент лобового сопротивления при нулевой подъемной силе. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *CX\_0 Calculate* и свяжите его с атрибутом *CX\_0*, т.е. перетащите скрипт на атрибут *CX\_0*.

- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *CX\_0 Calculate*, вычисляющего коэффициент лобового сопротивления при нулевой подъемной силе, являются атрибуты *\_M* и *Lamd\_fuz* (рис. 7).
- ◆
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *CX\_0 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.

```

begin
    Result := ( 0.882 + 0.147 * "_M" ) *
              ( 0.015288 + 0.000656 * "Lamd_fuz" +
                0.041 / sqr("Lamd_fuz") );
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.

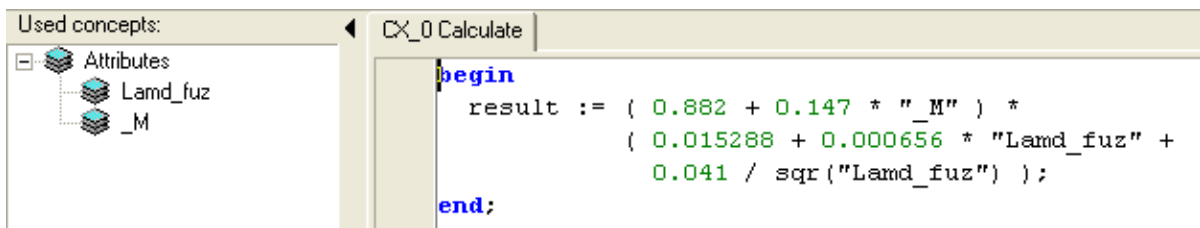


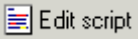
Рис. 7. Атрибуты и тело скрипта *CX\_0 Calculate*

◆

### 2.2.3.7 Определение скрипта для вычисления удельной нагрузки на крыло при заданной крейсерской скорости на расчетной высоте полета

Чтобы вычислить с помощью скрипта удельную нагрузку на крыло при заданной крейсерской скорости на расчетной высоте полета по формуле (6), необходимо выполнить следующие действия:


◆

- ◆ Создайте концепт «скрипт», который будет вычислять коэффициент лобового сопротивления при нулевой подъемной силе. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *PO\_H Calculate* и свяжите его с атрибутом *PO\_H*, т.е. перетащите скрипт на атрибут *PO\_H*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *PO\_H Calculate*, вычисляющего коэффициент лобового сопротивления при нулевой подъемной силе, являются атрибуты *V\_kr*, *DeltaH\_kr*, *CX\_0* и *Motn\_topl* (рис. 8).
- ◆
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *PO\_H 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.

```

begin
    Result := ( 0.208 * "DeltaH_kr" * sqr("V_kr")
                * sqrt ("CX_0") ) / (1 - 0.6 * "Motn_topl");
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.

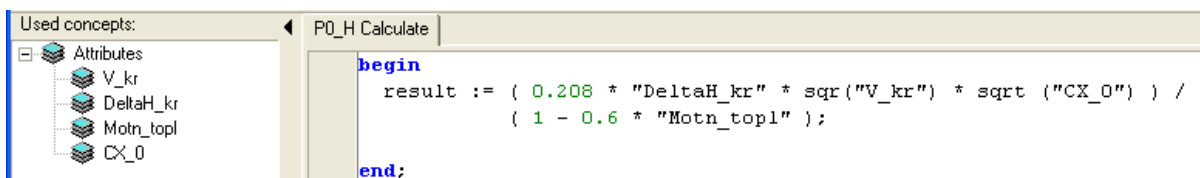



Рис. 8. Атрибуты и тело скрипта *PO\_H Calculate*


◆

### 2.2.3.8 Определение скрипта для вычисления удельной нагрузки на крыло

Чтобы вычислить с помощью скрипта удельную нагрузку на крыло по формуле (11), необходимо выполнить следующие действия:

- ◆
- ◆ Создайте концепт «скрипт», который будет вычислять коэффициент лобового сопротивления при нулевой подъемной силе. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *PO Calculate* и свяжите его с атрибутом *PO*, т.е. перетащите скрипт на атрибут *PO*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *PO Calculate*, вычисляющего коэффициент лобового сопротивления при нулевой подъемной силе, являются атрибуты *PO\_zahpos* и *PO\_H* (рис. 9).
- ◆
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *PO 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.

```
begin
    if "PO_zahpos" < "PO_H"
    then result := "PO_zahpos"
    else result := "PO_H";
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.

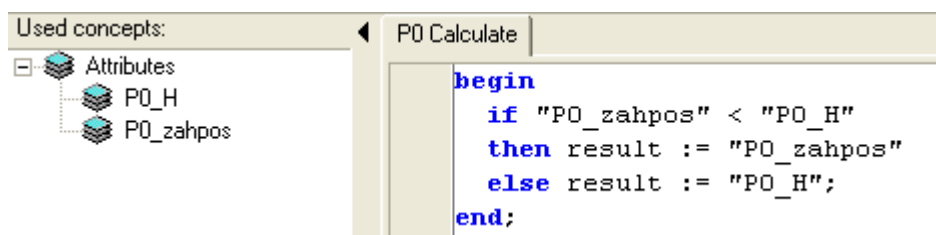


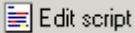
Рис. 9. Атрибуты и тело скрипта *PO Calculate*




### 2.2.3.9 Определение скрипта для вычисления максимального аэродинамического качества

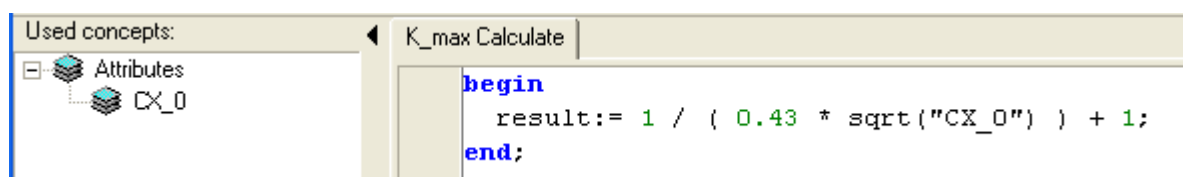
Чтобы вычислить с помощью скрипта максимальное аэродинамическое качество по формуле (10), необходимо выполнить следующие действия:



- ◆ Создайте концепт «скрипт», который будет вычислять максимальное аэродинамическое качество. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *K\_max Calculate* и свяжите его с атрибутом *K\_max*, т.е. перетащите скрипт на атрибут *K\_max*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *K\_max Calculate*, вычисляющего максимальное аэродинамическое качество, является атрибут *CX\_0* (рис. 10).
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *K\_max 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.

```
begin
    result := 1 / ( 0.43 * sqrt("CX_0") ) + 1;
end;
```

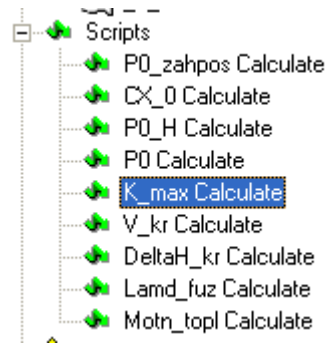
- ◆ Сохраните скрипт нажатием на кнопку . Закройте окно редактора скриптов.
- ◆ Проверьте правильность синтаксиса скрипта нажатием на кнопку *Check syntax*. Если будет зафиксирована ошибка, необходимо вызвать редактор скриптов и внести необходимые исправления.



**Рис. 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. 11).



**Рис. 11. Скрипты дескриптивной онтологии**

#### **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. 12).

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

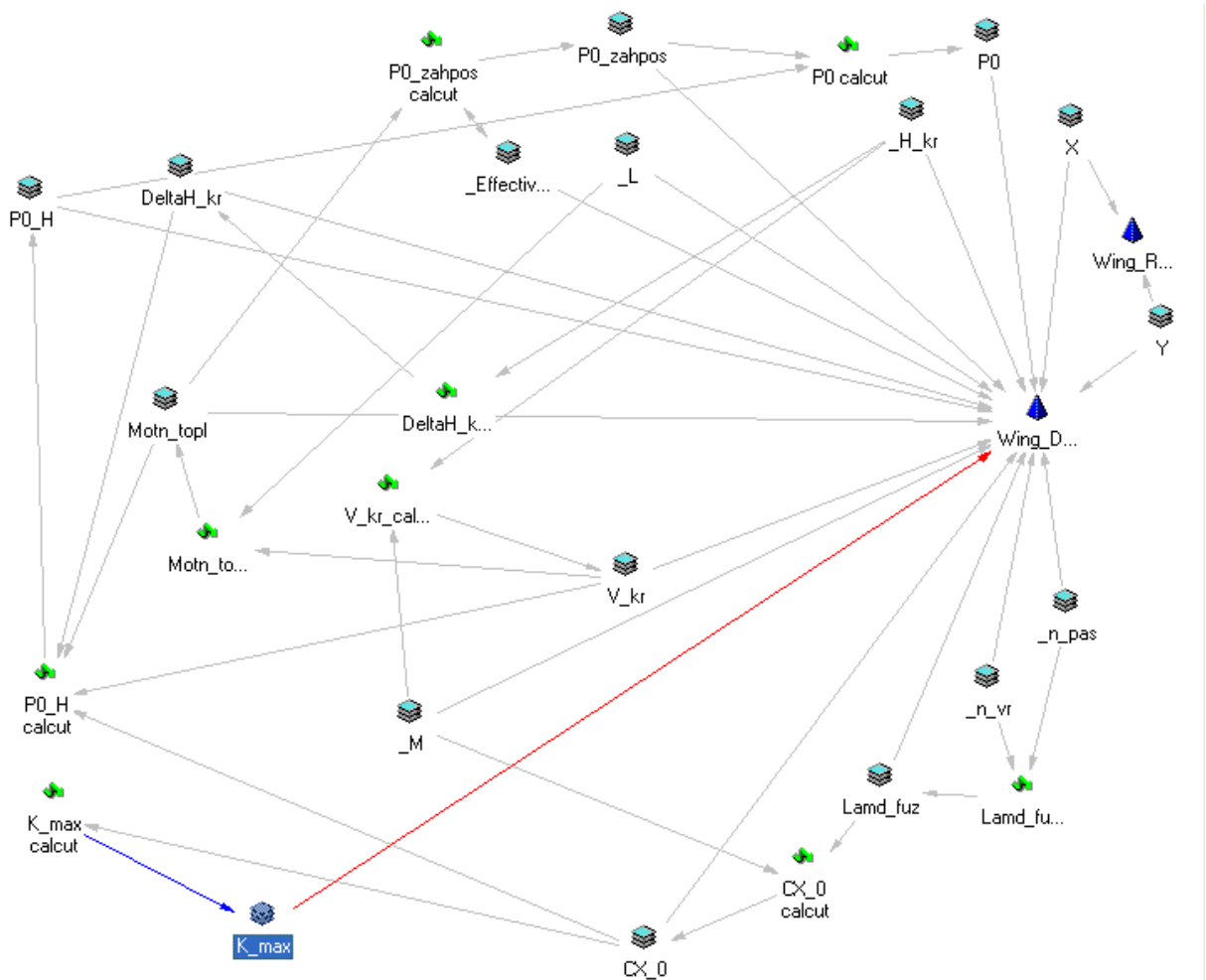


Рис. 12. Представление онтологии предметной области “Выбор удельной нагрузки на крыло” в виде семантической сети

## 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*. 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. 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. 13).

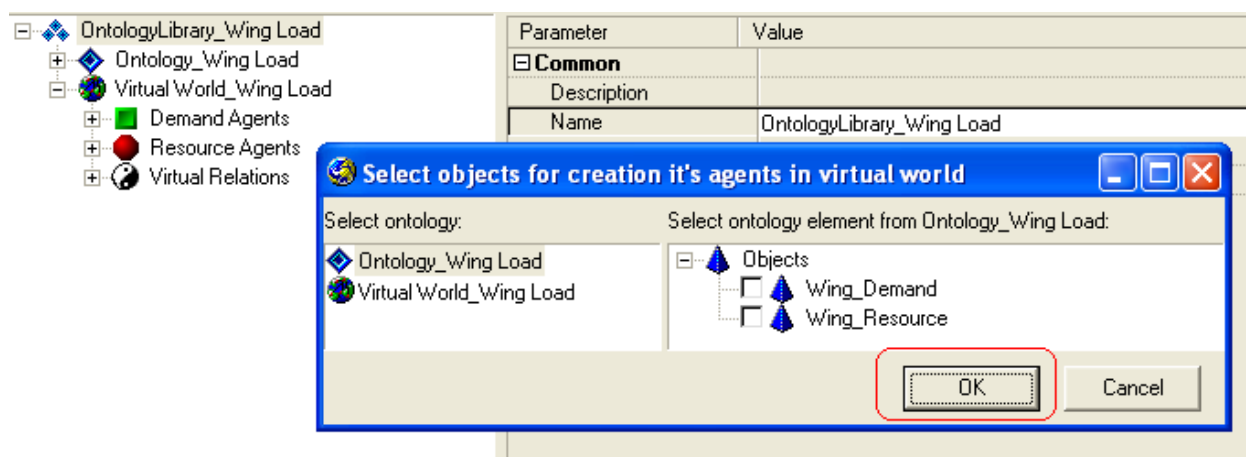


Рис. 13. Выбор объектов, для которых должны создаваться агенты

- ◆
- ◆ 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. 14).
- ◆

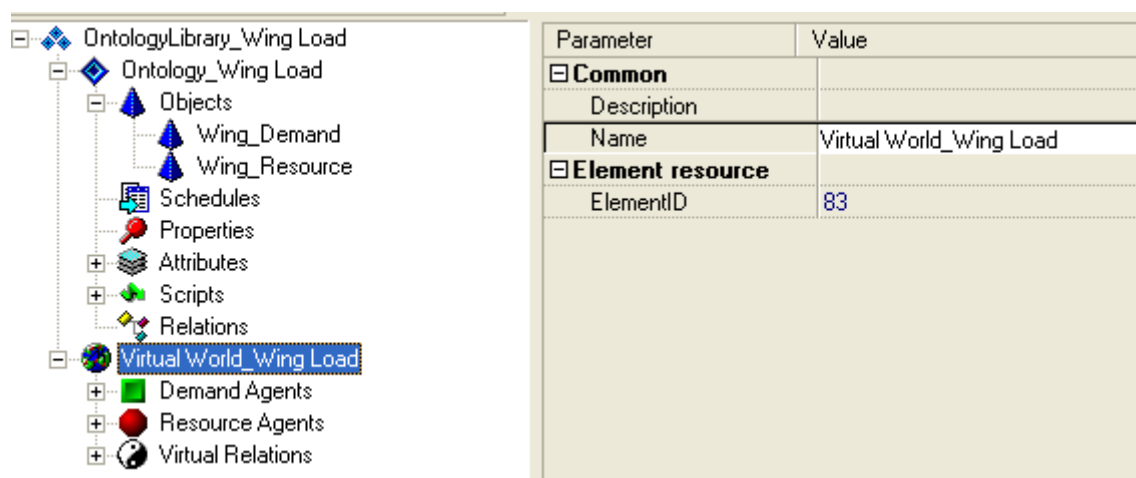


Рис. 14. Категории концептов онтологии мира заказов и ресурсов

### 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 *Wing\_Demand*. Затем нажмите *<OK>* и переименуйте созданный концепт в *Wing\_Demand* (рис. 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).
- ◆

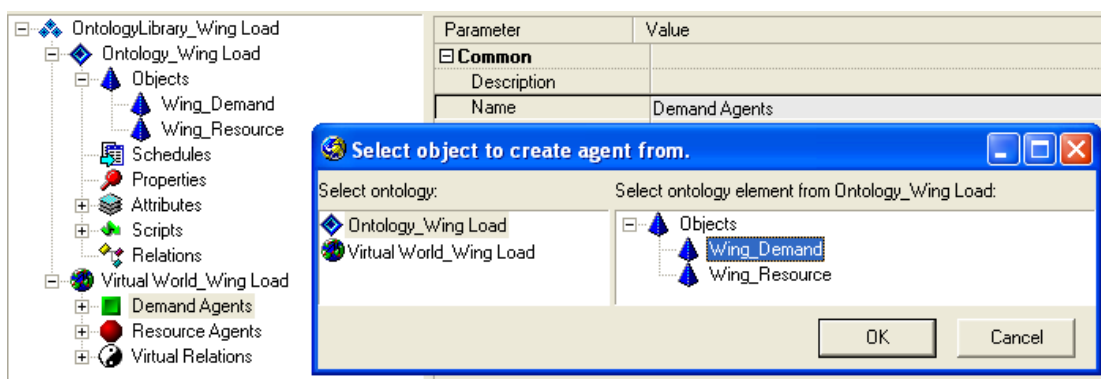


Рис. 15. Создание агента заказа для концепта *Wing\_Demand*

### 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 *BD\_Plane*. Затем нажмите *<OK>* и переименуйте созданный концепт в *BD\_Plane Resource* (рис. 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*.
- ◆

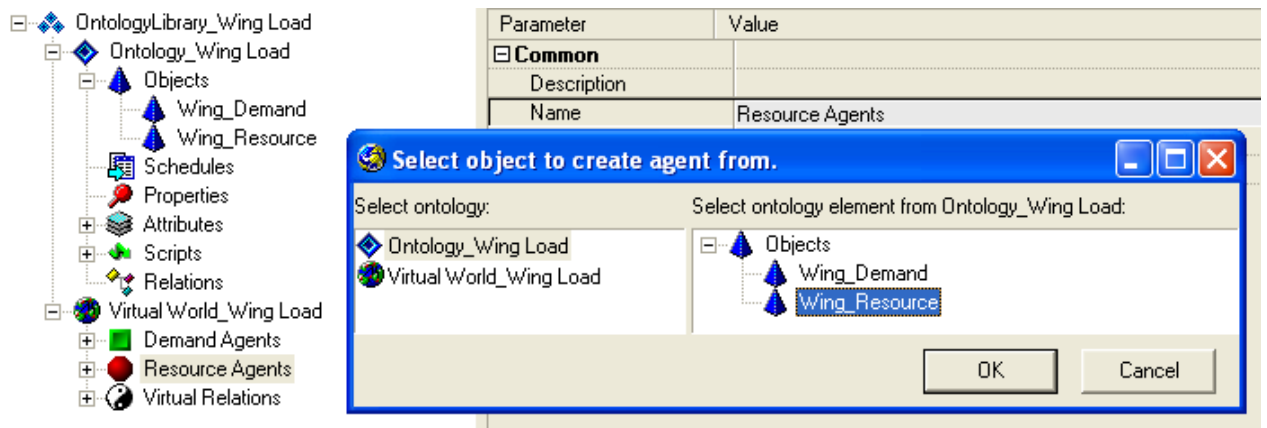


Рис. 16. Создание агента ресурса для концепта *Wing\_Resource*

### 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 *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. 17).
- ◆

Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Wing\_Demand.Wing\_Resource* (рис. 17).

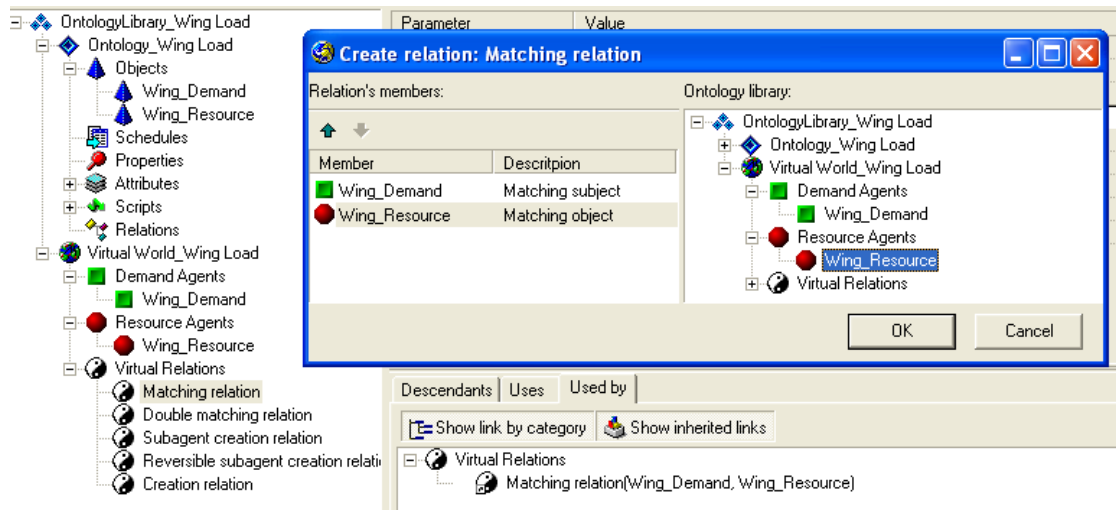


Рис. 17. Связывание отношением матчинга концептов *Wing\_Demand* и *Wing\_Resource*

### 2.3.4 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. 18). При этом откроется окно редактирования условий матчинга (рис. 19).
- ◆

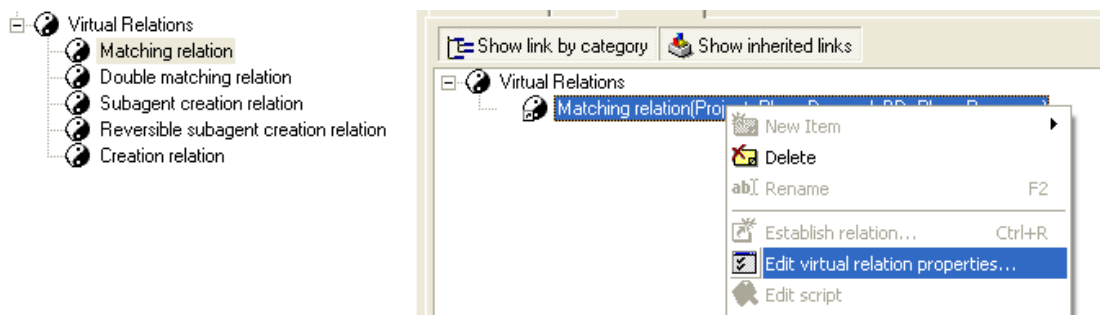
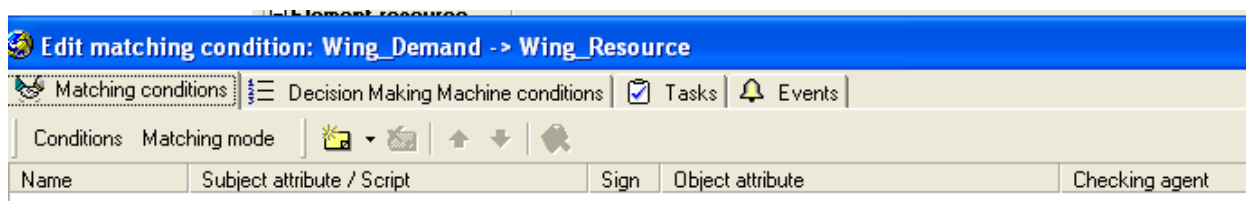


Рис. 18. Редактирование свойств отношения матчинга

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 матчере (в данной работе не используется).



**Рис. 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.

### 2.3.5 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.

#### 2.3.5.1 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 (11) следует принять минимальное из двух значений: удельной нагрузки на крыло при заходе на посадку (формула (1)) и удельной нагрузки на крыло при заданной крейсерской скорости на расчетной высоте полета (формула (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. 20):
  - Attribute = 'Wing\_Demand.P0';
  - Order = 'Min';
  - Weight = '100'.
- ◆ Активизируйте условие принятия решения, выставив флажок *Active*.
- ◆

### 2.3.5.2 Создание условия принятия решения - максимизация аэродинамического качества

- ◆
- ◆ Создайте условие 2 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. 20):
  - Attribute = 'Wing\_Demand.K\_max';
  - Order = 'Max';
  - Weight = '100'.
- ◆ Активизируйте условие принятия решения, выставив флажок *Active*.
- ◆

Условия принятия решения в матчинге для выполнения расчетов приведены на рис. 20.

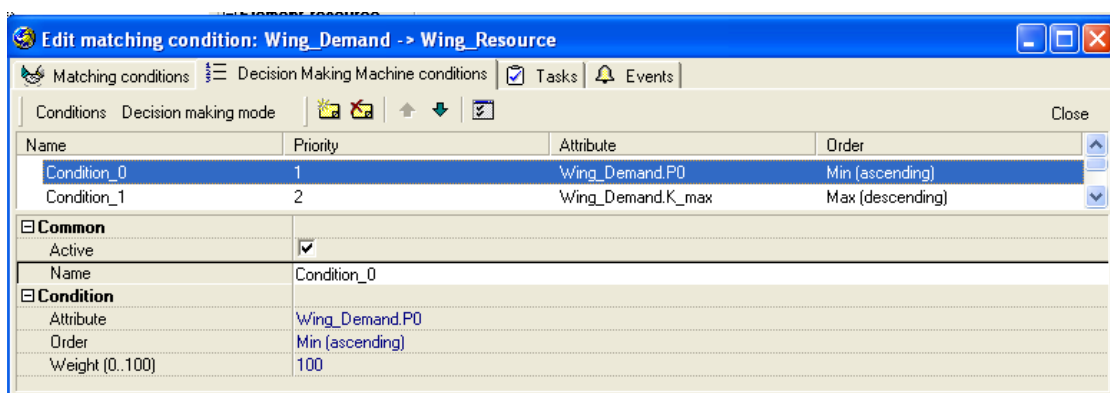


Рис. 20 – Условия принятия решения в матчинге *Wing\_Demand* – *Wing\_Resource*

### 2.3.6 Параметры представления экземпляра объекта в сцене

Поведение экземпляра концепта «объект» (агента) в сцене определяет Группа свойств концепта «объект» *Interface behaviour* (рис. 21).

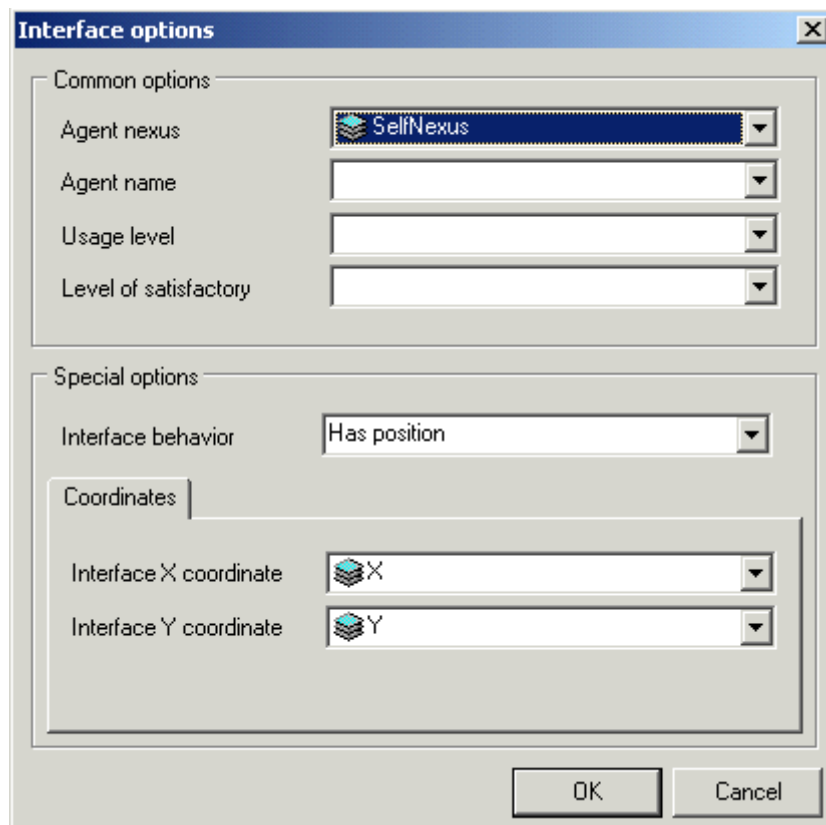


Рис. 21. Редактор свойств *Interface behaviour*

- ◆
- ◆ 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>.
- ◆

### 2.3.7 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. 22.

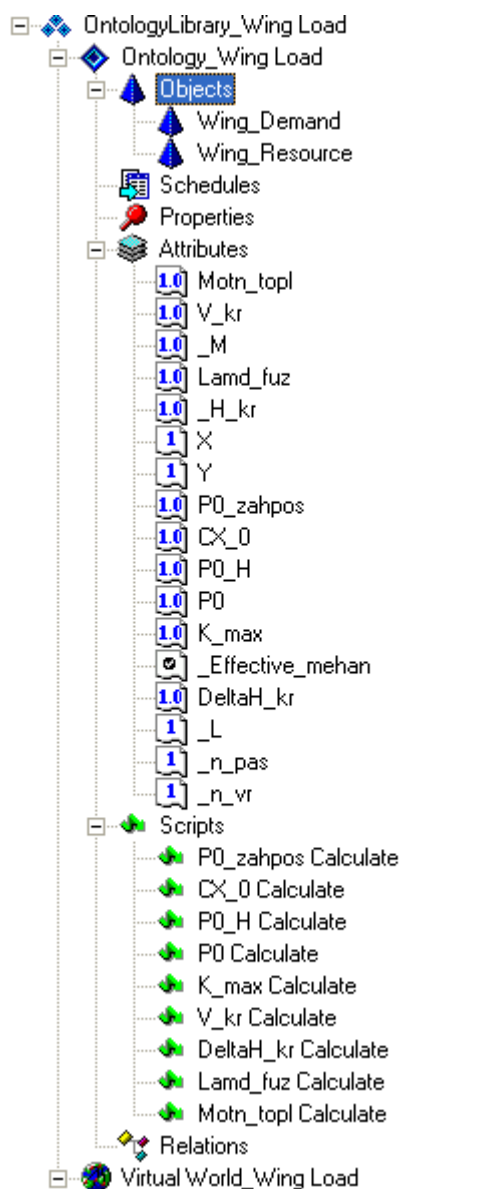



Рис. 22. Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор удельной нагрузки на крыло”

- ◆
- ◆ 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*).
- ◆



## 2.4 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. 23).

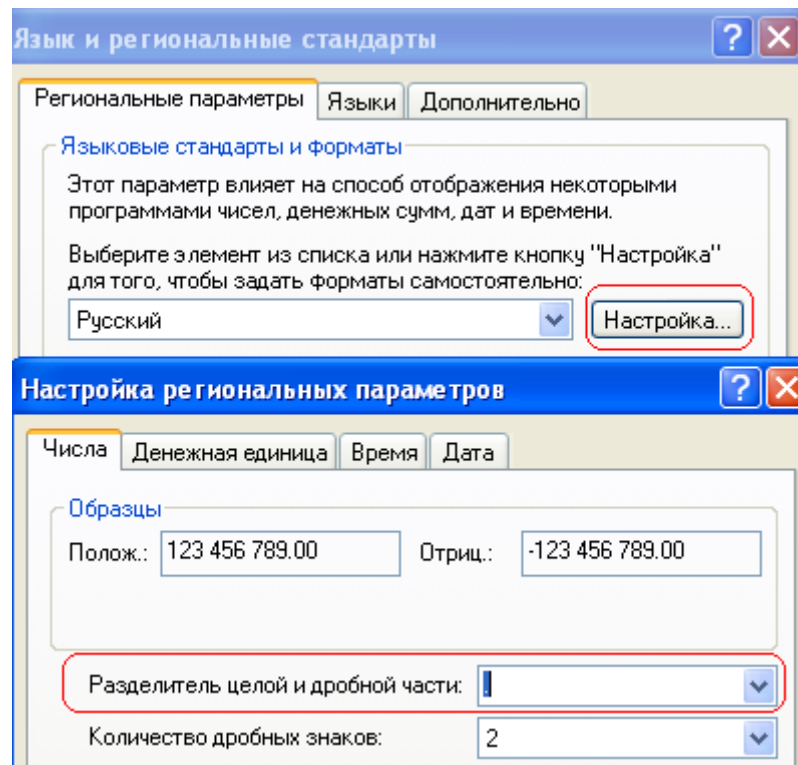



Рис. 23. Настройки компьютера

- ◆ Вызовите исполняющую систему, запуская программу  Unifntf.exe, находящуюся в папке *OntConsUniIntf*.
- ◆
- ◆ Создайте новую онтологическую сцену (*File → New scene -> Load ontology*, выберите онтологию *Wing Load ontology.ocf*).
- ◆
- ◆
- ◆ В окне физического мира создайте одного агента проектируемого самолета *Wing Demand\_1* и одного агента объекта-ресурса *Wing Resource\_1* для выполнения расчетов (рис. 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>_Effective_mehan</b>	+	+	+
<b>_H_kr</b>	10100	10000	8100
<b>_M</b>	0.88	0.88	0.47
<b>_L</b>	5200	4350	1350
<b>_n_pas</b>	164	350	32
<b>_n_vr(3..10)</b>	6	8	3

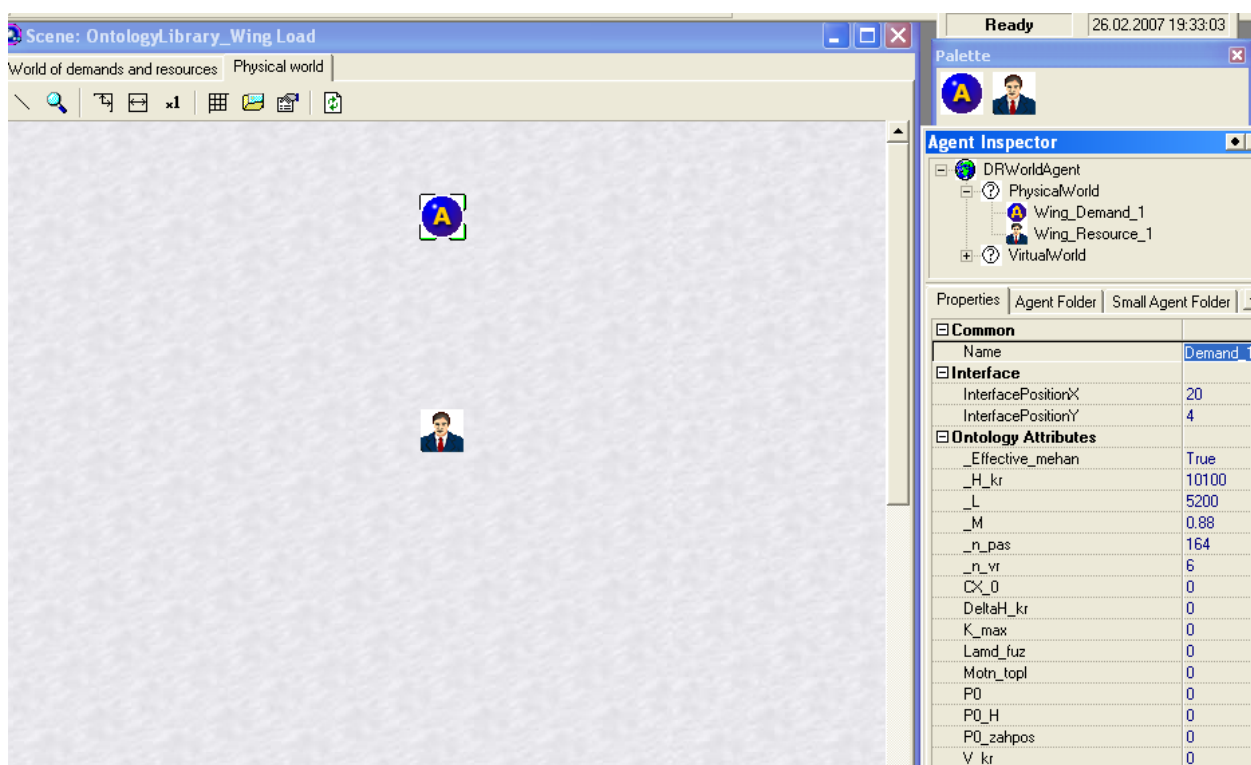



Рис. 24. Создание онтологической сцены предметной области «Выбор удельной нагрузки на крыло»

◆

## 2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА

### 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 *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_0	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

Рис. 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

Рис. 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.*

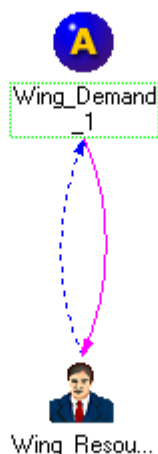




Рис. 27. Результаты матчинга

### 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\_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*).
- ◆

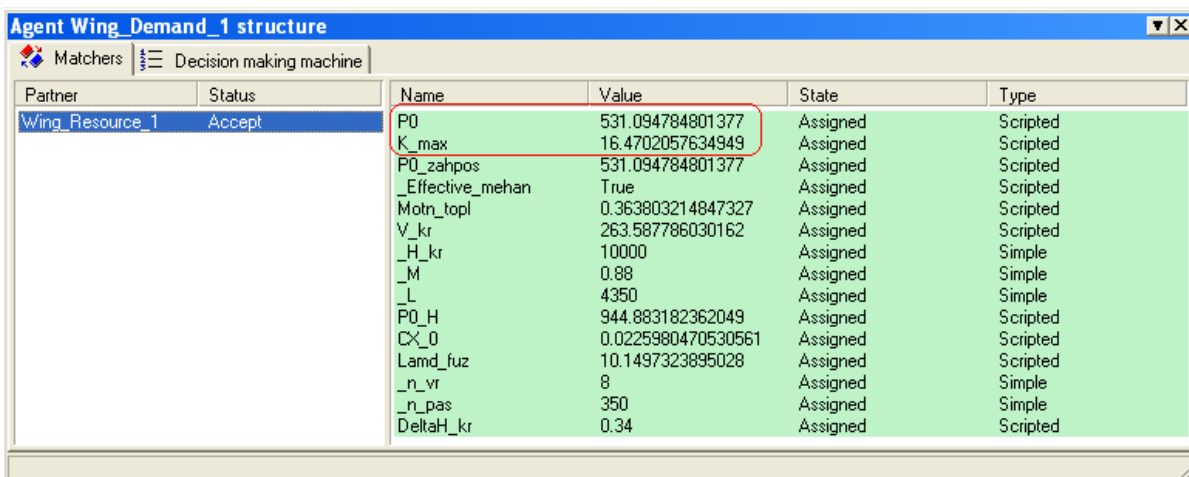
### 2.5.3. Загрузка ранее созданной сцены виртуального мира

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

- ◆
- ◆ Загрузите сцену *Scene\_Wing Load TY-154.osf*.
- ◆

## 2.5.4 Моделирование сцены для самолета ИЛ-86

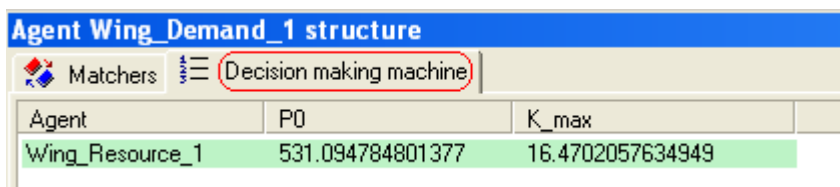
- ◆
- ◆ Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*).
- ◆ У агента проекта *Wing Demand\_1* измените значения атрибутов.
- ◆ Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*).
- ◆ При помощи кнопки  запустите сцену на моделирование (т.е., запустите процесс матчинга). Наблюдайте процесс матчинга между агентами проекта и самолетов-прототипов.
- ◆ Результаты матчинга показаны на рис. 28, а таблица принятия решения агента проекта – на рис. 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
		Motn_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

Рис. 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

Рис. 29. Таблица принятия решений агента *Wing Demand\_1* самолета ИЛ-86


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

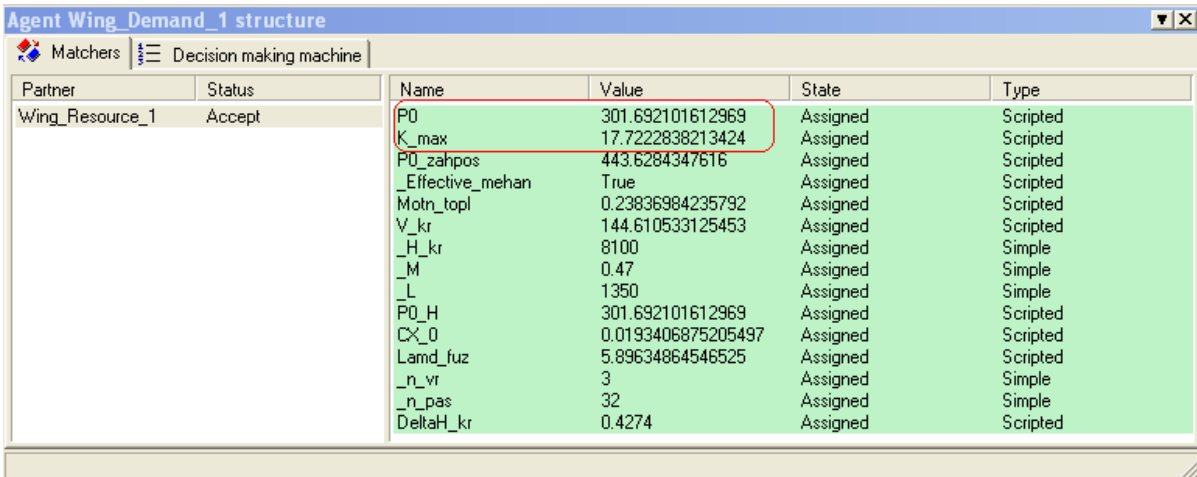
◆

- ◆ 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*).

### 2.5.5 Моделирование сцены для самолета ЯК-40

- ◆ При помощи средств онтологической загрузки загрузите сцену *Scene\_Wing Load TV-154.osf*.
- ◆ 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. 30, а таблица принятия решения агента проекта – на рис. 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
		Motr_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

Рис. 30. Структура агента проекта *Wing Demand\_1* самолета ЯК-40

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 31), в которой перечислены основные параметры, вычисляемые с помощью скриптов).

Agent	PO	K_max
Wing_Resource_1	301.692101612969	17.7222838213424

Рис. 31. Таблица принятия решений агента *Wing Demand\_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\_Wing Load ЯК-40*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*.
- ◆ 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 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.
11. Как установить или изменить значения атрибутов агентов с помощью Инспектора агентов?
12. Какие действия необходимо выполнить в процессе моделирования онтологической сцены?

## 5 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ