

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

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

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ  
ПРИ ВЫБОРЕ ПОТРЕБНОЙ  
ТЯГОВООРУЖЕННОСТИ САМОЛЕТА

*Утверждено Редакционно-издательским советом университета  
в качестве методических указаний к лабораторной работе № 3*

САМАРА  
Издательство СГАУ  
2008

УДК 629.13

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

Рецензент д-р техн. наук, проф. А.Н. Коварцев

**Использование онтологии при выборе потребной тяговооруженности самолета:** метод. указания к лаб. работе №3/ сост. Н.М. Боргест, Е.В. Симонова – Самара: Изд-во СГАУ, 2008. – 36 с.

В данной работе осуществлена разработка мультиагентного приложения, реализующего возможности использования скриптов для выполнения расчетов при выборе потребной тяговооруженности самолета, приведенной к взлетному режиму.

Предназначены для использования в учебном процессе специальности 220305 – Автоматизированное управление жизненным циклом продукции при изучении курса «Онтология производственной сферы».

## СОДЕРЖАНИЕ

1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ .....	4
2 ВЫБОР ЗНАЧЕНИЯ ПОТРЕБНОЙ ТЯГОВОООРУЖЕННОСТИ САМОЛЕТА, ПРИВЕДЕННОЙ К ВЗЛЕТНОМУ РЕЖИМУ .....	5
2.1. Постановка задачи .....	5
2.2 Проектирование дескриптивной онтологии .....	7
2.3 Проектирование онтологии мира заказов и ресурсов .....	21
2.4 Создание онтологической сцены .....	28
2.5 Моделирование сцены виртуального мира.....	30
3 КОНТРОЛЬНЫЕ ВОПРОСЫ .....	35
4 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ .....	35

## 1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

На основании нижеприведенных характеристик самолета необходимо рассчитать потребную тяговооруженность самолета, приведенную к взлетному режиму.

Характеристики самолетов-прототипов:

- эффективная механизация, да/нет;
- удельная нагрузка на крыло,  $daH/m^2$ ;
- длина разбега по ВПП, м;
- количество двигателей на самолете, шт.;
- максимальное аэродинамическое качество,
- число Маха;
- высота полета, м.

Основные характеристики самолетов-прототипов, а также требования к проектируемому самолету могут быть описаны в виде *онтологий*. 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 №3 – разработка мультиагентного приложения, реализующего возможности скриптов для расчетов в процессе одностороннего матчинга при принятии решения по выбору тяговооруженности самолета.

В процессе выполнения лабораторной работы №3 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 ЗНАЧЕНИЯ ПОТРЕБНОЙ ТЯГОВООРУЖЕННОСТИ САМОЛЕТА, ПРИВЕДЕННОЙ К ВЗЛЕТНОМУ РЕЖИМУ

### 2.1. Постановка задачи

2.1.1 Расчет тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости полета на расчетной высоте

Значение тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте полета определяется по следующей формуле:

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

где

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

Аэродинамическое качество на крейсерском режиме рассчитывается как:

$$K_{крейс} = 0,8 * K_{max}, \quad (2)$$

где

- $K_{max}$  – максимальное аэродинамическое качество (рассчитывается в лабораторной работе 2).

Относительная плотность воздуха на крейсерской высоте полета рассчитывается следующим образом:

$$\Delta H_{кр} = 0,8 - 0,000046 H_{кр}, \quad (3)$$

где

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

Коэффициент, учитывающий изменение тяги двигателя по скорости полета, рассчитывается как

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

где

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

### 2.1.2 Расчет тяговооруженности из условия заданной длины разбега

Тяговооруженность из условия обеспечения заданной длины пробега определяется следующим образом:

$$\overline{P_0^{II}} = \frac{1,26p_0}{C_{y \max \text{ взл}} \cdot L_{\text{разб}}} + 0,09975, \quad (5)$$

где

- $p_0$  – удельная нагрузка на крыло (рассчитывается в лабораторной работе 2),  $\text{даН/м}^2$ ;
- $L_{\text{разб}}$  – длина разбега по ВПП (исходное данное), м;
- $C_{y \max \text{ взл}}$  – значение максимального коэффициента подъемной силы самолета во взлетной конфигурации; значение  $C_{y \max \text{ взл}}$  выбирается в зависимости от системы механизации крыла:
  - для эффективной механизации  $C_{y \max \text{ взл}} = 2,5$ ;
  - для слабой (неэффективной) механизации  $C_{y \max \text{ взл}} = 2,0$ .

### 2.1.3 Расчет тяговооруженности из условия взлета при отказе одного двигателя

Тяговооруженность из условия взлета при отказе одного двигателя определяется в виде

$$\overline{P_0^{III}} = \frac{1,5 \cdot n_{\text{де}}}{n_{\text{де}} - 1} (0,083 + tg\theta_{\min}), \quad (6)$$

где

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

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

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

#### 2.1.4 Расчет потребной тяговооруженности самолета

Потребная тяговооруженность самолета, приведенная к взлетному режиму, имеет вид

$$P = \max (\overline{P}_0^I, \overline{P}_0^{II}, \overline{P}_0^{III}), \quad (7)$$

где

- $\overline{P}$  – расчетное значение потребной тяговооруженности ;
- $\overline{P}_0^I$  – тяговооружённость из условия обеспечения горизонтального полета на крейсерской скорости полета  $V_{\text{крейс}}$  на расчетной высоте  $H_{\text{крейс}}$  определяется по формуле (1);
- $\overline{P}_0^{II}$  – тяговооружённость из условия заданной длины разбега, рассчитывается по формуле (5);
- $\overline{P}_0^{III}$  – тяговооружённость из условия взлета при отказе одного двигателя, определяется 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\_Draught*.
- Создайте дескриптивную онтологию предметной области «Выбор потребной тяговооруженности самолета» (*New Item -> Descriptive ontology*). По умолчанию дескриптивная онтология имеет имя *Ontology\_1*. Переименуйте ее, вводя в поле *Name* имя *Ontology\_Draught*.
-

## 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 потребной тяговооруженности самолета» необходимо создать два концепта «объект»:

- *Demand Draught* (проектируемый самолет) с атрибутами
  - *Effective mehan* (эффективная механизация крыла: да/нет);
  - *P* (нагрузка на крыло);
  - *Dist Razb* (длина разбега по ВПП);
  - *N dvig* (количество двигателей);
  - *K\_max* (максимальное аэродинамическое качество);
  - *M* (число Маха);
  - *H\_kr* (высота полета);
  - *DeltaH\_kr* (относительная плотность воздуха на расчетной высоте);
  - *Ksi* (коэффициент, учитывающий изменение тяги двигателя по скорости полета);
  - *K\_kreis* (аэродинамическое качество на крейсерском режиме);
  - *TG* (градиент набора высоты на взлетном режиме);
  - *P0\_1* (тяговооруженность из условия обеспечения горизонтального полета на крейсерской скорости полета на расчетной высоте);
  - *P0\_2* (тяговооруженность из условия заданной длины разбега);
  - *P0\_3* (тяговооруженность из условия набора высоты при одном отказавшем двигателе);
  - *P0\_max* (расчетное значение потребной тяговооруженности);
  - *Temp* (вспомогательная переменная, используется в скриптах).
- *Resource Draught* – объект-ресурс, в данной задаче не имеет атрибутов, он необходим только для того, чтобы запустить матчинг, в процессе которого будут выполнены необходимые расчеты.

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



- 
- Создайте объект *Demand\_Draught* (*Objects* -> *New Item* -> *Object*), переименуйте его в *Demand\_Draught*, изменяя значение в поле *Name*, назначьте этому объекту пиктограммы, соответствующие трем видам отображения.
- Создайте объект *Resource\_Draught* (аналогично созданию *Demand\_Draught*), переименуйте его в *Resource\_Draught*, 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 вещественный атрибут *H\_kr* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *H\_kr*.
- Создайте вещественный атрибут *M* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *M*.
- Создайте вещественный атрибут *P* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *P*.
- Создайте вещественный атрибут *Dist\_razb* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *Dist\_razb*.
- Создайте целый атрибут *N\_dvig* (*New Item* -> *Integer Attribute*), переименуйте созданный атрибут в *N\_dvig*.
- Создайте вещественный атрибут *K\_max* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *K\_max*.
- Создайте вещественный атрибут *DeltaH\_kr* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *DeltaH\_kr*.
- Создайте вещественный атрибут *Ksi* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *Ksi*.
- Создайте вещественный атрибут *K\_kreis* (*New Item* -> *Float Attribute*), переименуйте созданный атрибут в *K\_kreis*.
- Создайте вещественный атрибут *TG* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *TG*.
- Создайте вещественный атрибут *P0\_1* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *P0\_1*.
- Создайте вещественный атрибут *P0\_2* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *P0\_2*.

- Создайте вещественный атрибут *P0\_3* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *P0\_3*.
- Создайте вещественный атрибут *P0\_max* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *P0\_max*.
- Создайте вещественный атрибут *Temp* (*Attributes* -> *New Item* -> *Float Attribute*), переименуйте созданный атрибут в *Temp*.
- 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 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 потребной тяговооруженности самолета», т.е. указать, что объект *Demand Draught* имеет атрибуты *Effective mehan*, *H kr*, *M*, *P*, *Dist razb*, *N dvig*, *K max*, *DeltaH kr*, *Ksi*, *K kreis*, *TG*, *P0\_1*, *P0\_2*, *P0\_3*, *P0\_max*, *Temp*, *X*, *Y*, а объект *Resource Draught* – атрибуты *X,Y*, необходимо в дереве концептов «перетащить» атрибуты *Effective mehan*, *H kr*, *M*, *P*, *Dist razb*, *N dvig*, *K max*, *DeltaH kr*, *Ksi*, *K kreis*, *TG*, *P0\_1*, *P0\_2*, *P0\_3*, *P0\_max*, *Temp*, *X*, *Y* на объект *Demand Draught*. Аналогично следует «перетащить» атрибуты *X,Y* на объект *Resource Draught*.

- 
- Установите связи между атрибутами *Effective mehan*, *H kr*, *M*, *P*, *Dist razb*, *N dvig*, *K max*, *DeltaH kr*, *Ksi*, *K kreis*, *TG*, *P0\_1*, *P0\_2*, *P0\_3*, *P0\_max*, *Temp*, *X*, *Y* и объектом *Demand Draught*.
- Установите связи между атрибутами *X,Y* и объектом *Resource Draught*.
- 

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 *Demand Draught* в закладке *Uses*, причем данный объект имеет атрибуты *Effective mehan*, *H kr*, *M*, *P*, *Dist razb*, *N dvig*, *K max*, *DeltaH kr*, *Ksi*, *K kreis*, *TG*, *P0\_1*, *P0\_2*, *P0\_3*, *P0\_max*, *Temp*, *X*, *Y* (соответствующие связи отображены в закладке *Uses*). В закладке *Used by* атрибутов отображена их связь с соответствующим объектом.

The image shows a software interface with two main panels. The left panel is a tree view of an ontology hierarchy. The right panel shows the 'Uses' tab for the 'Demand\_Draught' object, including a 'Parameter' table and a list of attributes.

**Parameter Table:**

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

**Attributes List:**



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

Рис. 1. Атрибуты (свойства) объекта *Demand\_Draught* 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 с 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 аэродинамического качества на крейсерском режиме

Чтобы вычислить с помощью скрипта аэродинамическое качество на крейсерском режиме по формуле (2), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять аэродинамическое качество на крейсерском режиме. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item* -> *Script*.
- Созданный скрипт переименуйте в *K\_kreis Calculate* и свяжите его с атрибутом *K\_kreis*, т.е. перетащите скрипт на атрибут *K\_kreis*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметром скрипта *K\_kreis Calculate*, вычисляющего аэродинамическое качество на крейсерском режиме, является атрибут *\_K\_max* (рис. 2).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *K\_kreis Calculate* в дереве концептов дескриптивной онтологии, затем перейдите в закладку *Script body*, нажмите на кнопку . При этом откроется окно редактора скриптов, в котором необходимо набрать следующий текст (имена концептов заключаются в кавычки, незначащие пробелы в начале идентификаторов не допускаются). Имена концептов необходимо выбирать в списке концептов, которые являются параметрами скрипта. Скрипт заканчивается точкой с запятой.

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

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

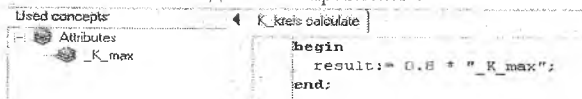




Рис. 2. Атрибуты и тело скрипта *K\_kreis Calculate*

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

Чтобы вычислить с помощью скрипта относительную плотность топлива на расчетной высоте по формуле (3), 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*). Параметром скрипта *DeltaH\_kr Calculate*, вычисляющего относительную плотность воздуха на расчетной высоте, является атрибут *\_H\_kr* (рис. 3).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *DeltaH\_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.

```
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.

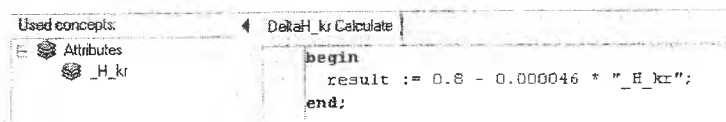




Рис. 3. Атрибуты и тело скрипта *DeltaH\_kr Calculate*

### 2.2.3.3 Определение скрипта для вычисления коэффициента, учитывающего изменение тяги двигателя по скорости полета

Чтобы вычислить с помощью скрипта коэффициент, учитывающий изменение тяги двигателя по скорости полета по формуле (4), необходимо выполнить следующие действия:

- Создайте концепт «скрипт», который будет вычислять коэффициент, учитывающий изменение тяги двигателя по скорости полета. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *Ksi Calculate* и свяжите его с атрибутом *Ksi*, т.е. перетащите скрипт на атрибут *Ksi*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметром скрипта *Ksi Calculate*, вычисляющего коэффициент, учитывающий изменение тяги двигателя по скорости полета, является атрибут *\_M* (рис. 4).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *Ksi 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 - 0.32 * "_M" + 0.4 * sqr("_M")
            - 0.01 * sqr("_M") * "_M";
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.

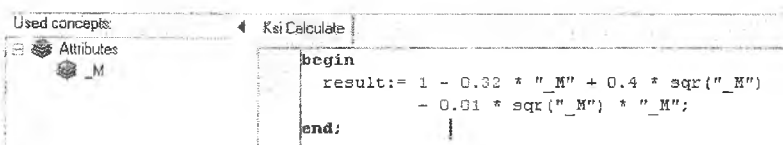



Рис. 4. Атрибуты и тело скрипта *Ksi Calculate*

➤

### 2.2.3.4 Определение скрипта для вычисления тяговооруженности из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте

Чтобы вычислить с помощью скрипта тяговооруженность из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте по формуле (1), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять тяговооруженность из условия обеспечения горизонтального полета на крейсерской скорости 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 *P0\_1 Calculate* и свяжите его с атрибутом *P0\_1*, т.е. перетащите скрипт на атрибут *P0\_1*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *P0\_1 Calculate*, вычисляющего тяговооруженность из условия обеспечения горизонтального полета на крейсерской скорости на расчетной высоте, являются атрибуты *Ksi*, *K\_kreis* и *DeltaH\_kr* (рис. 5).
- Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *P0\_1 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 / ( "Ksi" * exp( 0.85*ln("DeltaH_kr") )
        * 0.85 * "K_kreis");
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 *P0\_1 Calculate*


➤

### 2.2.3.5 Определение скрипта для вычисления тяговооруженности из условия обеспечения заданной длины разбега

Чтобы вычислить с помощью скрипта тяговооруженность из условия обеспечения заданной длины разбега по формуле (5), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять тяговооруженность из условия обеспечения заданной длины разбега. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *PO\_2 Calculate* и свяжите его с атрибутом *PO\_2*, т.е. перетащите скрипт на атрибут *PO\_2*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *PO\_2 Calculate*, вычисляющего тяговооруженность из условия заданной длины разбега, являются атрибуты *\_P*, *Dist\_razb*, *Effective\_mehan* (рис. 6).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *PO\_2 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 "_Effective_mehan"
    then result:= 1.26*_P/( 2.5 * "_Dist_Razb" )+ 0.09975
    else result:= 1.26*_P/(2.0 * "_Dist_Razb" )+ 0.09975;
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.

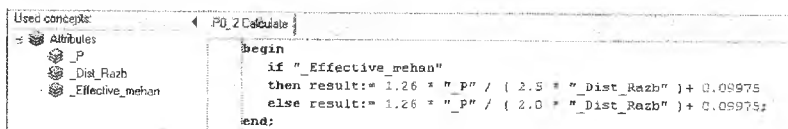



Рис. 6. Атрибуты и тело скрипта *PO\_2 Calculate*




### 2.2.3.6 Определение скрипта для вычисления градиента набора высоты при взлетном режиме

Чтобы вычислить с помощью скрипта градиент набора высоты при взлетном режиме, необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять градиент набора высоты при взлетном режиме. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *TG Calculate* и свяжите его с атрибутом *TG*, т.е. перетащите скрипт на атрибут *TG*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *TG Calculate*, вычисляющего градиент набора высоты при взлетном режиме, является атрибут *\_N\_dvig* (рис. 7).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *TG 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 "_N_dvig" = 2 then result:= 0.024
    else if "_N_dvig" = 3 then result:= 0.027
    else result:= 0.03;
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.

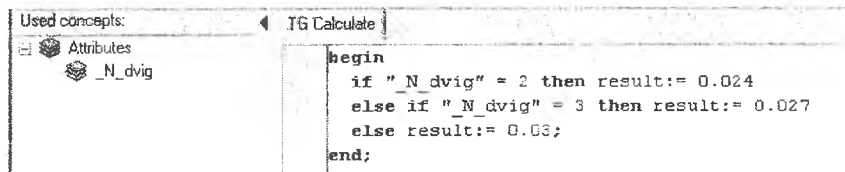




Рис. 7. Атрибуты и тело скрипта *TG Calculate*

### 2.2.3.7 Определение скрипта для вычисления тяговооруженности из условия набора высоты при одном отказавшем двигателе

Чтобы вычислить с помощью скрипта тяговооруженность из условия набора высоты при одном отказавшем двигателе по формуле (6), необходимо выполнить следующие действия:

- Создайте концепт «скрипт», который будет вычислять тяговооруженность из условия набора высоты при одном отказавшем двигателе. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *P0\_3 Calculate* и свяжите его с атрибутом *P0\_3*, т.е. перетащите скрипт на атрибут *P0\_3*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *P0\_3 Calculate*, вычисляющего тяговооруженность из условия набора высоты при одном отказавшем двигателе, являются атрибуты *\_N\_dvig* и *TG* (рис. 8).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *P0\_3 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:= ( (1.5*_N_dvig) / (_N_dvig"-1) )
              * ( 0.083+"TG" );
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.

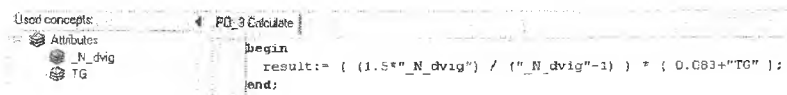




Рис. 8. Атрибуты и тело скрипта *P0\_3 Calculate*

### 2.2.3.8 Определение скрипта для вычисления потребной тяговооруженности

Чтобы вычислить с помощью скрипта потребную тяговооруженность по формуле (7), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять потребную тяговооруженность. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *PO\_max Calculate* и свяжите его с атрибутом *PO\_max*, т.е. перетащите скрипт на атрибут *PO\_max*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *PO\_max Calculate*, вычисляющего потребную тяговооруженность, являются атрибуты *PO\_1*, *PO\_2*, *PO\_3* и вспомогательная переменная *Temp* (рис. 9).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *PO\_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
    "Temp" := "PO_1";
    if "Temp" < "PO_2" then "Temp" := "PO_2";
    if "Temp" < "PO_3" then "Temp" := "PO_3";
    result := "Temp";
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.

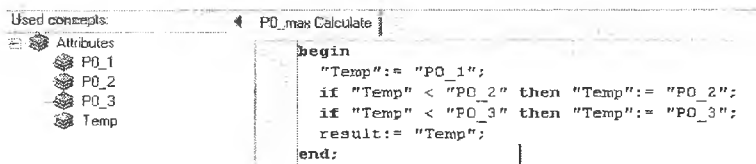


Рис. 9. Атрибуты и тело скрипта *PO\_max Calculate*

- 
- Таким образом, в дереве концептов дескриптивной онтологии предметной области “Выбор потребной тяговооруженности самолета” можно видеть следующие скрипты (рис. 10).

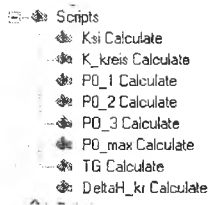


Рис. 10. Скрипты дескриптивной онтологии

#### 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. 11).

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

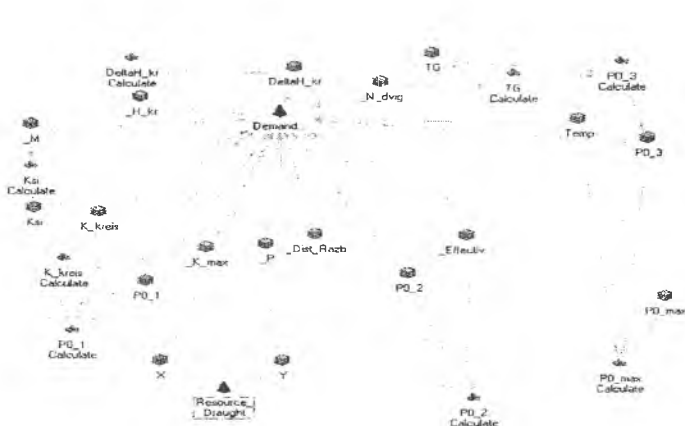


Рис. 11. Представление онтологии предметной области “Выбор потребной тяговооруженности самолета” в виде семантической сети

## 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. 12).

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. 12).

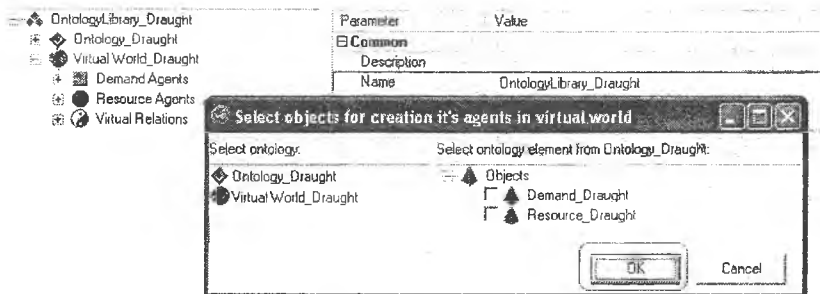


Рис. 12. Выбор объектов, для которых должны создаваться агенты

- Создайте онтологию мира заказов и ресурсов для предметной области «Выбор удельной нагрузки на крыло» (*New Item -> Virtual World Ontology*). Введите в поле *Name* название онтологии – *Virtual World Draught*.
- Раскройте дерево концептов онтологии виртуального мира (рис. 13).

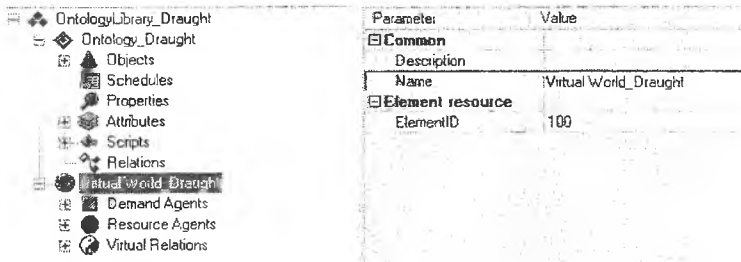


Рис. 13. Категории концептов онтологии мира заказов и ресурсов

### 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 *Demand Draught*. Затем нажмите <OK> и переименуйте созданный концепт в *Demand Draught* (рис. 14).
- 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).
- 

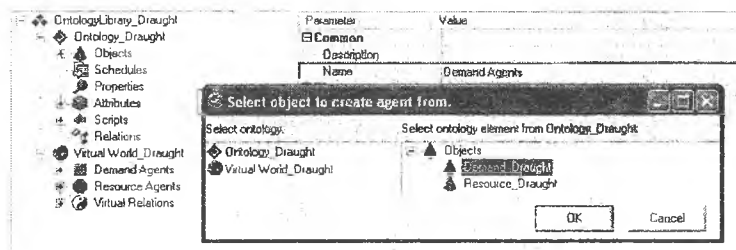


Рис. 14. Создание агента заказа для концепта *Demand Draught*

### 2.3.3 Создание концепта «агент ресурса»

- Создайте концепт «агент ресурса» для концепта «объект-ресурс»: выделите категорию *Resource Agents*, в контекстном меню выберите пункт *New Item -> Resource agent* и в появившемся диалоге выберите концепт

- *Resource\_Draught*. Затем нажмите <OK> и переименуйте созданный концепт в *Resource\_Draught* (рис. 15).
- 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 *raoAutoCreate*.
- Не устанавливайте у концепта флаг *raoActive*.
- 

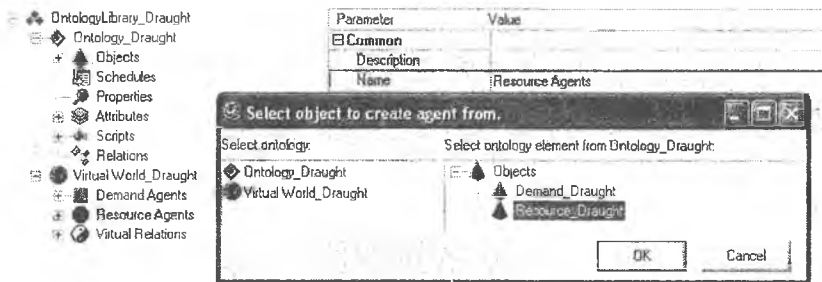


Рис. 15. Создание агента ресурса для концепта *Resource\_Draught*

### 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 необязательно состоится: агенты могут не договориться по разным причинам (есть более выгодное предложение, данное предложение не устраивает партнера/агента и т.д.).

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

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

*ResourceAgents*. В качестве *Matching subject* выберите *Demand\_Draught*, а в качестве *Matching object* – *Resource\_Draught* (рис. 16).

Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Demand\_Draught*. *Resource\_Draught* (рис. 16).

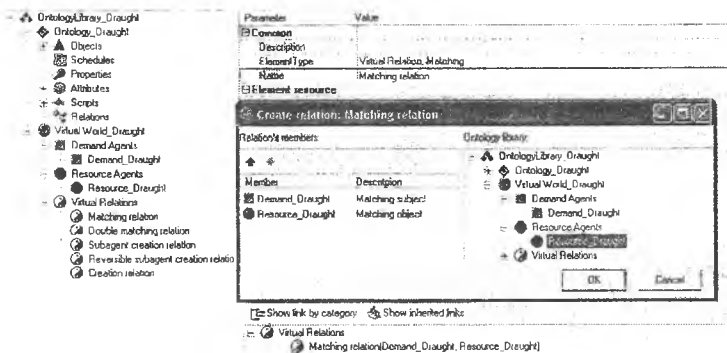


Рис. 16. Связывание отношением матчинга концептов *Demand\_Draught* и *Resource\_Draught*

### 2.3.5 Условия матчинга

- 
- Перейдите в закладку *Used by* (*Virtual relations* -> *Matching relation* -> *Used by*) и выделите *Matching relation* (*Demand\_Draught*, *Resource\_Draught*), далее в контекстном меню выберите *Edit virtual relation properties* (рис. 17). При этом откроется окно редактирования условий матчинга (рис. 18).
- 

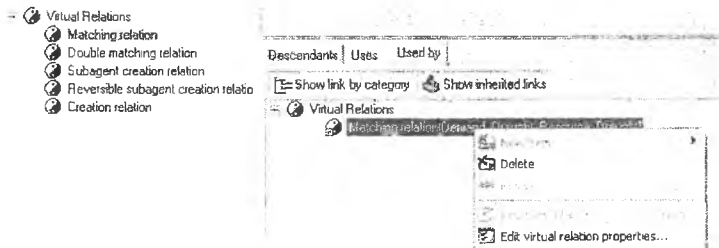


Рис. 17. Редактирование свойств отношения матчинга



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).

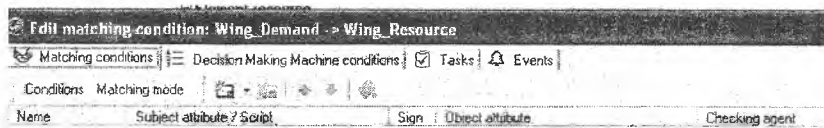



Рис. 18. Окно редактирования условий матчинга

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 максимального значения тяговооруженности

За расчетное значение потребной тяговооруженности самолета, согласно формуле (7), следует принять максимальное из трех значений: тяговооруженности из условия горизонтального полета на крейсерской

скорости (формула (1)), тяговооруженности из условия заданной длины разбега (формула(5)) и тяговооруженности из условия набора высоты при одном отказавшем двигателе (формула (6)).

- 
- Создайте условие 1 принятия решений для матчнга *Wing\_Demand* -> *Wing\_Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия (рис. 19):
  - Attribute = 'Demand\_Draught.P0\_max';
  - Order = 'Max';
  - Weight = '100'.
- Активизируйте условие принятия решения, выставив флажок *Active*.
- 

Условие принятия решения в матчнге для выполнения расчетов приведено на рис. 19.

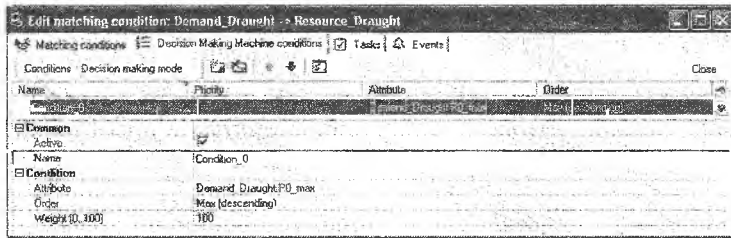


Рис. 19. Условие принятия решения в матчнге *Demand\_Draught* – *Resource\_Draught*

### 2.3.7 Параметры представления экземпляра объекта в сцене

Поведение экземпляра концепта «объект» (агента) в сцене определяет группа свойств концепта «объект» *Interface behaviour* (рис. 20).

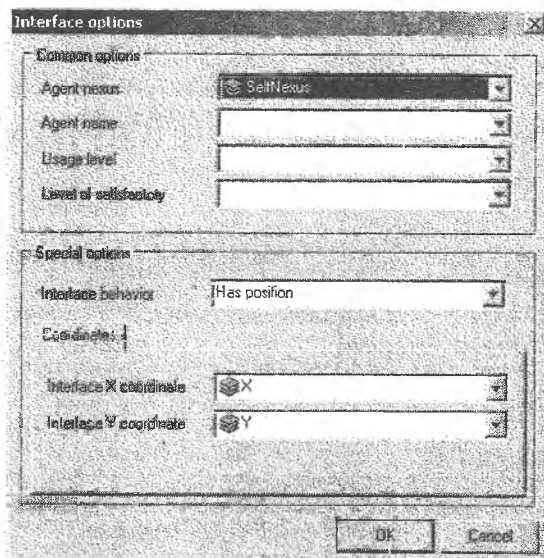


Рис. 20. Редактор свойств *Interface behaviour*

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

### 2.3.8 Сохранение онтологий предметной области “Выбор потребной тяговооруженности самолета”

Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор потребной тяговооруженности самолета” приведено на рис. 21.

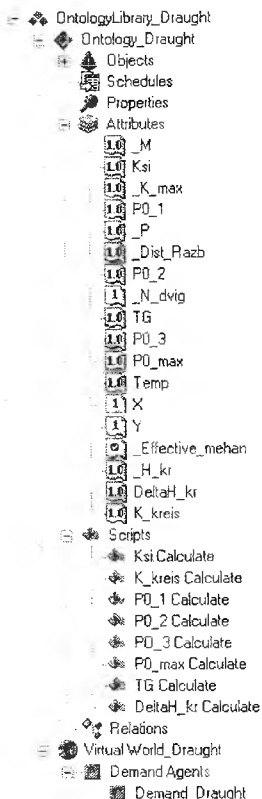



Рис. 21. Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор потребной тяговооруженности самолета”

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

## 2.4 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. 22).

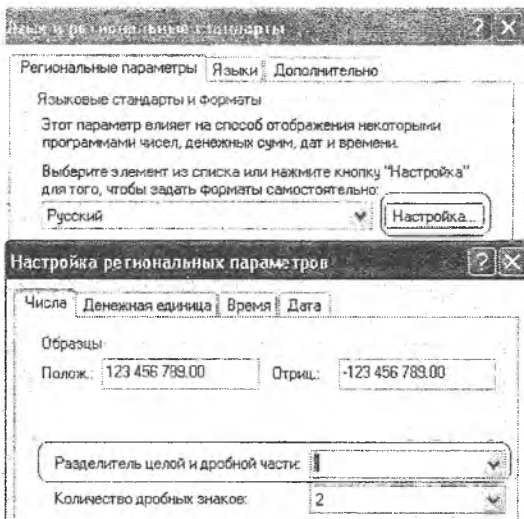



Рис. 22. Настройки компьютера

- Вызовите исполняющую систему, запуская программу  `UniIntf.exe`, находящуюся в папке `OntConsUniIntf`.
- 
- Создайте новую онтологическую сцену (`File ? New scene -> Load ontology`, выберите онтологию `Ontology_Draught.ocl`).
- 
- В окне физического мира создайте одного агента проектируемого самолета `Demand_Draught_1` и одного агента объекта-ресурса `Resource_Draught_1` для выполнения расчетов (рис. 23). С помощью инспектора агентов установите следующие значения атрибутов для агентов, например, для самолета Ту-154.

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

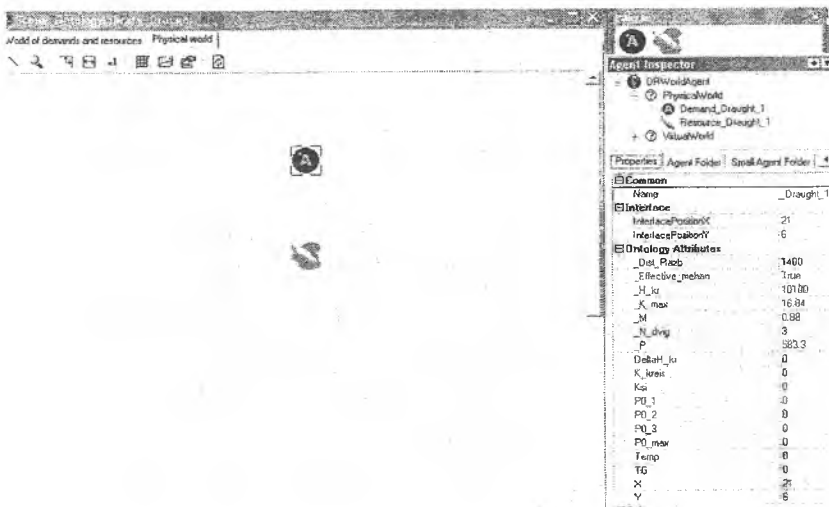



Рис. 23. Создание онтологической сцены предметной области «Выбор потребности втяговооруженности самолета»



## 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 *Demand\_Draught\_1* в окне физического или виртуального мира (либо при выполнении в контекстном меню агента команд *Actions* -> *Show agent structure*) можно открыть таблицу, в которой представлена структура агента (в закладке *Matchers*). В таблице указывается список ресурсов (партнеров), т.е. агентов, которые зарезервированы агентом-проектом (рис. 24). При выделении партнера в правом окне отображаются атрибуты агента-проекта (простые (*Simple*),

значения которых задавались непосредственно в сцене, и вычисляемые (*Scripted*), значения которых рассчитывались в процессе матчинга с помощью скриптов).

Partner	Status	Name	Value	State	Type
Resource_Draught_1	Accepted	PO_max	0.309738	Assigned	Scripted
		temp	0	Assigned	Simple
		PO_3	9.2476	Assigned	Scripted
		TG	0.027	Assigned	Simple
		NI_avg	0	Assigned	Scripted
		PO_2	0.309738	Assigned	Simple
		Effective_mehan	True	Assigned	Simple
		_Dist_Rado	1400	Assigned	Simple
		_P	593.0	Assigned	Simple
		PO_1	0.215395235275956	Assigned	Scripted
		Distah_kr	0.854	Assigned	Scripted
		_H_kr	10100	Assigned	Simple
		K_kolei	13.422	Assigned	Scripted
		K_max	16.84	Assigned	Simple
		Ksi	1.62134928	Assigned	Scripted
		_M	0.88	Assigned	Simple

Рис. 24. Структура агента проекта *Demand\_Draught\_1* самолета Ту-154

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 25), в которой перечислены основные параметры, вычисляемые с помощью скриптов.

Agent	PO_max
Resource_Draught_1	0.309738

Рис. 25. Таблица принятия решений агента *Demand\_Draught\_1* самолета Ту-154

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


Окончательные результаты матчинга можно видеть на рис. 26. В результате матчинга была выполнена следующая операция резервирования:

- *Demand\_Draught\_1 – Resource\_Draught\_1*.




Рис. 26. 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\_Draught\_Ty-154*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*.
- Завершите работу с исполняющей системой (*File* -> *Close*).
- 

## 2.5.3. Загрузка ранее созданной сцены виртуального мира

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

- 
- Загрузите сцену *Scene\_Draught\_Ty-154.osf*.
- 

## 2.5.4 Моделирование сцены для самолета Ил-86

- 
- Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools* -> *Clear results*).
- У агента проекта *Demand\_Draught\_1* измените значения атрибутов для самолета Ил-86.
- Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*).
- При помощи кнопки  запустите сцену на моделирование (т.е. запустите процесс матчинга). Наблюдайте процесс матчинга между агентами проекта и самолетов-прототипов.
- Результаты матчинга показаны на рис. 27, а таблица принятия решения агента проекта – на рис. 28.
-



Partner	Status	Name	Value	State	Type
Resource_Draught_1	Accept	PO_max	0.2670465	Assigned	Scripted
		Temp	0	Assigned	Scripted
		PO_3	0.226	Assigned	Scripted
		TG	0.08	Assigned	Scripted
		_N_dvag	4	Assigned	Simple
		PO_2	0.2670465	Assigned	Scripted
		_Effective_mehen	False	Assigned	Simple
		_Dist_Pzad	2000	Assigned	Simple
		_P	591.1	Assigned	Simple
		PO_1	0.218703687970413	Assigned	Scripted
		DvzokH_kz	0.34	Assigned	Scripted
		_H_jr	70000	Assigned	Simple
		K_kreis	0.176	Assigned	Scripted
		K_max	16.47	Assigned	Simple
		Ka	1.02134526	Assigned	Scripted
		_M	0.86	Assigned	Simple

Рис. 27. Структура агента проекта *Demand\_Draught\_1* самолета Ил-86

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проекта (рис. 28), в которой перечислены основные параметры, вычисляемые с помощью скриптов.

Agent Demand_Draught_1 structure	
Agent	PO_max
Resource_Draught_1	0.2670465


Рис. 28. Таблица принятия решений агента *Demand\_Draught\_1* самолета Ил-86

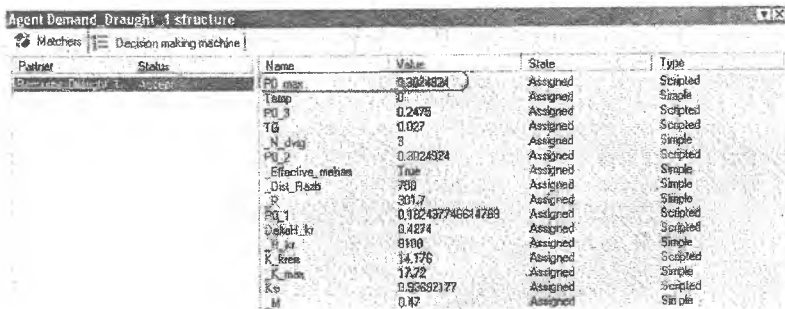
Агент-ресурс в данной задаче не имеет атрибутов, поэтому таблица, содержащая его структуру, пуста.

- 
- При помощи средств онтологического сохранения сохраните сцену под именем *Scene\_Draught\_Ил-86*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*.
- Завершите работу с исполняющей системой (*File -> Close*).
- 

### 2.5.5 Моделирование сцены для самолета Як-40

- 
- При помощи средств онтологической загрузки загрузите сцену *Scene\_Draught\_Ту-154.osf*.
- Выполните очистку окна виртуального мира, стирая результаты моделирования (*Tools -> Clear results*).

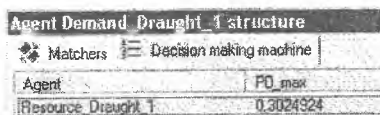
- У агента проекта *Demand\_Draught\_1* измените значения атрибутов для самолета Як-40.
- Перейдите в окно виртуального мира (в закладку *Worlds of demands and resources*).
- При помощи кнопки  запустите сцену на моделирование (т.е. запустите процесс матчинга). Наблюдайте процесс матчинга между агентами проекта и самолетов-прототипов.
- Результаты матчинга показаны на рис. 29, а таблица принятия решения агента проекта – на рис. 30.
- 



Partner	Status	Name	Value	State	Type
Demand_Draught_1	Active	PO_max	0.3024324	Assigned	Scripted
		Temp	0	Assigned	Simple
		PO_3	0.2478	Assigned	Scripted
		TG	0.027	Assigned	Scripted
		N_diviz	3	Assigned	Simple
		PO_2	0.3024324	Assigned	Scripted
		Effective_mehan	True	Assigned	Simple
		Dist_Rash	750	Assigned	Simple
		R	301.7	Assigned	Simple
		PO_1	0.182437746614789	Assigned	Scripted
		Dokht_kr	0.4274	Assigned	Scripted
		H_jir	8100	Assigned	Simple
		K_kreis	14.176	Assigned	Scripted
		K_max	17.72	Assigned	Simple
		Kp	0.55692177	Assigned	Scripted
		M	0.47	Assigned	Simple

Рис. 29. Структура агента проекта *Demand\_Draught\_1* самолета Як-40

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



Agent	PO_max
Resource_Draught_1	0.3024324

Рис. 30. Таблица принятия решений агента *Demand\_Draught\_1* Як-40

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

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

### 3 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?

### 4 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

*Учебное издание*

**ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ ПРИ ВЫБОРЕ  
ПОТРЕБНОЙ ТЯГОВООРУЖЕННОСТИ САМОЛЕТА**

*Методические указания*

Составители.

***Боргест Николай Михайлович  
Симонова Елена Витальевна***

Редактор Т.И. Кузнецова  
Компьютерная доверстка И.И. Спиридонова

Подписано в печать 10.12.2008 г. Формат 60x84 1/16.

Бумага офсетная. Печать офсетная.

Печ.л. 2,25.

Тираж 100 экз. Заказ **140**. Арт. С-45/2008.

Самарский государственный аэрокосмический университет.  
443086 Самара, Московское шоссе, 34.

Издательство Самарского государственного  
аэрокосмического университета.  
443086 Самара, Московское шоссе, 34.