

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

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

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ  
ПРИ ВЫБОРЕ ДВИГАТЕЛЯ  
ДЛЯ ПРОЕКТИРУЕМОГО САМОЛЕТА

*Утверждено Редакционно-издательским советом университета  
в качестве методических указаний к лабораторной работе № 4*

САМАРА  
Издательство СГАУ  
2008

Составители: Н.М. Боргест, Е.В. Симонова  
Рецензент д-р техн. наук, проф. А.Н. Коварцев

**Использование онтологии при выборе двигателя для проектируемого самолета:** метод. указания к лаб. работе №4 / сост. Н.М. Боргест, Е.В. Симонова – Самара: Изд-во СГАУ, 2008. – 36 с.

В данной работе осуществлена разработка мультиагентного приложения, реализующего возможности использования онтологии в процессе одностороннего матчинга при выборе двигателя для проектируемого самолета.

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 ДВИГАТЕЛЯ ДЛЯ ПРОЕКТИРУЕМОГО САМОЛЕТА .....	4
2.1. Постановка задачи .....	4
2.2 Проектирование дескриптивной онтологии .....	7
2.3 Проектирование онтологии мира заказов и ресурсов .....	19
2.4 Создание онтологической сцены .....	28
2.5 Моделирование сцены виртуального мира.....	30
3 КОНТРОЛЬНЫЕ ВОПРОСЫ .....	32
4 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....	33

## 1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

На основании данных технического задания и параметров, рассчитанных в лабораторных работах №2 и №3, необходимо выбрать в базе данных двигатель, характеристики которого удовлетворяют требуемым условиям.

Основные характеристики двигателей-прототипов, а также требования к двигателю проектируемого самолета могут быть описаны в виде *онтологии*. При этом двигатель проектируемого самолета следует рассматривать как *проект* или *заказ*, а двигатели-прототипы, характеристики которых указаны в базе данных, – как *ресурсы*. На основании результатов процесса поиска взаимного соответствия между заказом и ресурсами (*матчинга*) принимаются или пересматриваются решения о бронировании или освобождении ресурсов (т.е. устанавливаются связи между заказом и адекватными ему ресурсами). Тем самым выполняется выбор двигателей-прототипов, чьи характеристики отвечают требованиям к двигателю проектируемого самолета.

Цель лабораторной работы № 4 – разработка мультиагентных приложений, реализующих возможности одностороннего матчинга в процессе принятия решений.

В процессе выполнения лабораторной работы № 4 решаются следующие задачи:

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

В начале каждого раздела излагаются общие теоретические положения. Идентификаторы концептов, атрибутов и скриптов записываются английским шрифтом. Действия, которые непосредственно предлагается выполнить пользователю, выделяются знаками, показанными ниже. Будьте, пожалуйста, внимательны!



## 2 ВЫБОР ДВИГАТЕЛЯ ДЛЯ ПРОЕКТИРУЕМОГО САМОЛЕТА

### 2.1. Постановка задачи

На основании заданных требований к двигателю проектируемого самолета по тяге, удельному расходу топлива и массе, а также на основании данных технического задания и параметров, рассчитанных в лабораторных

работах №2 и №3, необходимо выбрать в базе данных соответствующий двигатель.

Данные технического задания:

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

Принятые (выбранные значения):

- число двигателей на самолете, шт.

Расчетные данные:

- $\bar{m}_T$  – относительная масса топлива самолета, вычисляется по формуле (4) в лабораторной работе №2;
- $K_{\max}$  – максимальное аэродинамическое качество самолета, вычисляется по формуле (10) в лабораторной работе №2;
- $V_{кр}$  – крейсерская скорость самолета, вычисляется по формуле (5) в лабораторной работе №2;
- $\bar{P}_0$  – тяговооруженность самолета, вычисляется в лабораторной работе №3.

Необходимо выбрать в базе данных прототипы двигателей, отвечающие требованиям по расчетной потребной тяге двигателя, расчетному потребному удельному расходу топлива двигателя на крейсерском режиме и расчетной потребной массе двигателя. Среди этих прототипов необходимо выбрать двигатель, имеющий минимальную массу и обеспечивающий минимальный расход топлива на крейсерском режиме.

### 2.1.1 Расчет взлетной массы самолета первого приближения

Взлетная масса самолета первого приближения рассчитывается следующим образом:

$$m_0 = (m_{кн} + m_{сл}) / (1 - \bar{m}_T - \bar{m}_{пл} - \bar{m}_{сy}), \quad (1)$$

где

- $m_0$  – взлетная масса самолета, кг;
- $m_{кн}$  – масса коммерческой нагрузки, кг;
- $m_{сл}$  – масса служебной нагрузки, условно выбирается равной 300 кг;
- $\bar{m}_T$  – относительная масса топлива самолета, кг, вычисляется по формуле (4) в лабораторной работе №2;
- $\bar{m}_{пл}$  – относительная масса планера самолета без силовой установки, кг;
- $\bar{m}_{сy}$  – относительная масса силовой установки, кг.

Далее приведены максимально упрощенные формулы расчета этих величин.

Масса коммерческой нагрузки вычисляется как

$$m_{кн} = 115 * n_{пасс} \quad (2)$$

где

- $n_{пасс}$  – число пассажиров, чел (исходное данное).

Относительная масса планера самолета без силовой установки

$$\bar{m}_{пл} = 0,44 - 0,0003 * n_{пасс}, \quad (3)$$

где

- $n_{пасс}$  -- число пассажиров, чел (исходное данное).

Относительная масса силовой установки

$$\bar{m}_{сy} = 0,14 - 0,0002 * n_{пасс}, \quad (4)$$

где

- $n_{пасс}$  -- число пассажиров, чел (исходное данное).

### 2.1.2 Расчет минимальной потребной взлетной тяги двигателя

Минимальная потребная взлетная тяга двигателя определяется как

$$P_0 = \bar{P}_0 * m_0 / n_{дв}, \text{ кгс} \quad (5)$$

где

- $\bar{P}_0$  -- тяговооруженность самолета, рассчитывается по формуле (7) в лабораторной работе №3;
- $m_0$  -- взлетная масса самолета, кг, рассчитывается по формуле (1);
- $n_{дв}$  -- число двигателей на самолете (исходное данное).

### 2.1.3 Расчет максимального потребного (допустимого) удельного расхода топлива двигателя на крейсерском режиме

Максимальный потребный удельный расход топлива двигателя на крейсерском режиме

$$C_{уд\text{кр}} = 2,8 * \bar{m}_T * K_{max} * V_{кр} / (L_{пол} * (1 - 0,5 * \bar{m}_T) * (1,5 - 0,00003 * L_{пол})), \quad (6)$$

где

- $C_{уд\text{кр}}$  -- максимальный потребный удельный расход топлива двигателя на крейсерском режиме, кг топл/ кгс\*час;
- $\bar{m}_T$  -- относительная масса топлива самолета, вычисляется по формуле (4) в лабораторной работе №2;
- $K_{max}$  -- максимальное аэродинамическое качество самолета, рассчитывается по формуле (10) в лабораторной работе №2;
- $V_{кр}$  -- крейсерская скорость самолета, м/с, рассчитывается по формуле (5) в лабораторной работе №2;
- $L_{пол}$  -- дальность полета самолета, км (исходное данное).

## 2.1.4 Расчет максимальной потребной (допустимой) массы двигателя

Максимальная потребная масса двигателя определяется следующим образом:

$$m_{\text{дв}} = 0.8 * m_0 * \bar{m}_{\text{сy}} / n_{\text{дв}}, \quad (7)$$

где

- $m_0$  – взлетная масса самолета, кг, рассчитывается по формуле (1);
- $\bar{m}_{\text{сy}}$  – относительная масса силовой установки, рассчитывается по формуле (4);
- $n_{\text{дв}}$  – число двигателей на самолете (исходное данное).

## 2.2 Проектирование дескриптивной онтологии

### 2.2.1 Создание онтологин

- 
- 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\_Engine*.
- 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\_Engine*.
- 

### 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 «двигателя» необходимо создать два концепта «объект»:

- *Engine\_Demand* (двигатель проектируемого самолета) с атрибутами
  - *\_L* (дальность полета);
  - *\_N\_pas* (число пассажиров);
  - *\_N\_dvig* (число двигателей);

- $\underline{Motn\_topl}$  (относительная масса топлива);
- $\underline{K\_max}$  (максимальное аэродинамическое качество);
- $\underline{V\_kr}$  (крейсерская скорость полета);
- $\underline{P0}$  (тяговооруженность самолета);
- $\underline{M\_0}$  (взлетная масса самолета);
- $\underline{M\_kn}$  (масса коммерческой нагрузки);
- $\underline{M\_pl}$  (относительная масса планера самолета без силовой установки);
- $\underline{M\_su}$  (относительная масса силовой установки);
- $\underline{P\_0}$  (потребная взлетная тяга двигателя);
- $\underline{C\_kr}$  (удельный расход топлива двигателя на крейсерском режиме);
- $\underline{M\_dvig}$  (масса двигателя).

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

- *Engine\_Resource* (двигатель самолета-прототипа, имеющегося в базе данных) с атрибутами
  - *BD\_EngineName* (название двигателя, имеющегося в БД);
  - *BD\_P\_0* (тяга двигателя, имеющегося в БД);
  - *BD\_C\_kr* (удельный расход топлива двигателя, имеющегося в БД);
  - *BD\_M\_dvig* (масса двигателя, имеющегося в БД);

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

- 
- Создайте объект *Engine\_Demand* (*Objects -> New Item -> Object*), переименуйте его в *Engine\_Demand*, изменяя значение в поле *Name*, назначьте этому объекту пиктограммы, соответствующие трем видам отображения.
- Создайте объект *Engine\_Resource* (аналогично созданию *Wing\_Demand*), переименуйте его в *Engine\_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 целый атрибут *\_L* (*Attributes -> New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_L*.
- Создайте целый атрибут *\_N\_pas* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_N\_pas*.
- Создайте целый атрибут *\_N\_dvig* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *\_N\_dvig*.



- Создайте вещественный атрибут *\_Motn\_topl* (*New Item -> Float Attribute*), переименуйте созданный атрибут в *\_Motn\_topl*.
- Создайте вещественный атрибут *\_K\_max* (*New Item -> Float Attribute*), переименуйте созданный атрибут в *\_K\_max*.
- Создайте вещественный атрибут *\_V\_kr* (*New Item -> Float Attribute*), переименуйте созданный атрибут в *\_V\_kr*.
- Создайте вещественный атрибут *\_P0* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *\_P0*.
- Создайте целый атрибут *M\_0* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *M\_0*.
- Создайте целый атрибут *M\_kn* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *M\_kn*.
- Создайте вещественный атрибут *M\_pl* (*New Item -> Float Attribute*), переименуйте созданный атрибут в *M\_pl*.
- Создайте вещественный атрибут *M\_su* (*New Item -> Float Attribute*), переименуйте созданный атрибут в *M\_su*.
- Создайте целый атрибут *P\_0* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *P\_0*.
- Создайте вещественный атрибут *C\_kr* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *C\_kr*.
- Создайте целый атрибут *M\_dvig* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *M\_dvig*.
- 
- Создайте строковый атрибут *BD\_EngineName* (*Attributes -> New Item -> String Attribute*), переименуйте созданный атрибут в *BD\_EngineName*.
- Создайте целый атрибут *BD\_P\_0* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *BD\_P\_0*.
- Создайте вещественный атрибут *BD\_C\_kr* (*Attributes -> New Item -> Float Attribute*), переименуйте созданный атрибут в *BD\_C\_kr*.
- Создайте целый атрибут *BD\_M\_dvig* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *BD\_M\_dvig*.
- 
- 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 двигателя самолета», т.е. указать, что объект *Engine\_Demand* имеет атрибуты *\_L*, *\_N\_pas*, *\_N\_dvig*, *\_Motn\_topl*,

K\_max, V\_kr, P\_0, M\_0, M\_kn, M\_pl, M\_su, P\_0, C\_kr, M\_dvig, X, Y, а объект Engine\_Resource – атрибуты BD\_EngineName, BD\_P\_0, BD\_C\_kr, BD\_M\_dvig, X, Y, необходимо в дереве концептов «перетянуть» атрибуты L, N\_pas, N\_dvig, Motn\_topl, K\_max, V\_kr, P\_0, M\_0, M\_kn, M\_pl, M\_su, P\_0, C\_kr, M\_dvig, X, Y на объект Engine\_Demand. Аналогично следует «перетянуть» атрибуты BD\_EngineName, BD\_P\_0, BD\_C\_kr, BD\_M\_dvig, X, Y на объект Engine\_Resource.

- 
- Установите связи между атрибутами L, N\_pas, N\_dvig, Motn\_topl, K\_max, V\_kr, P\_0, M\_0, M\_kn, M\_pl, M\_su, P\_0, C\_kr, M\_dvig, X, Y и объектом Engine\_Demand.
- Установите связи между атрибутами BD\_EngineName, BD\_P\_0, BD\_C\_kr, BD\_M\_dvig, X, Y и объектом Engine\_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 Engine\_Demand в закладке Uses, причем данный объект имеет атрибуты L, N\_pas, N\_dvig, Motn\_topl, K\_max, V\_kr, P\_0, M\_0, M\_kn, M\_pl, M\_su, P\_0, C\_kr, M\_dvig, X, Y (соответствующие связи отображены в закладке Uses). В закладке Used by атрибутов отображена их связь с соответствующим объектом.

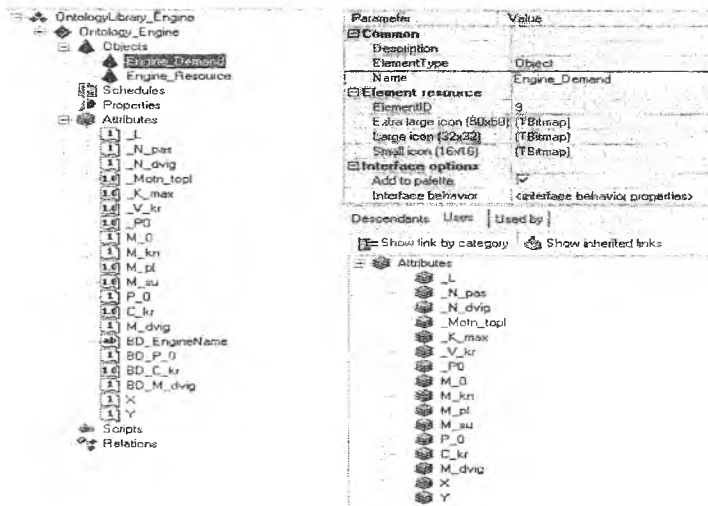




Рис. 1. Атрибуты (свойства) объекта Engine\_Demand в закладке Uses

### 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  *Edit script* в закладке *Script body*).

#### 2.2.3.1 Определение скрипта для вычисления массы коммерческой нагрузки


Чтобы вычислить с помощью скрипта массу коммерческой нагрузки по формуле (2), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять массу коммерческой нагрузки. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *M\_kn Calculate* и свяжите его с атрибутом *M\_kn*, т.е. перетащите скрипт на атрибут *M\_kn*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметром скрипта *M\_kn Calculate*, вычисляющего массу коммерческой нагрузки, является атрибут *\_N\_pas* (рис. 2).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *M\_kn 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 (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:= 115 * "_N_pas";
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.

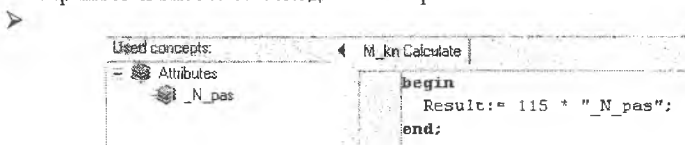



Рис. 2. Атрибуты и тело скрипта *M\_kn Calculate*

➤

### 2.2.3.2 Определение скрипта для вычисления относительной массы планера самолета без силовой установки


Чтобы вычислить с помощью скрипта относительную массу планера самолета без силовой установки по формуле (3), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять относительную массу планера самолета без силовой установки. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *M\_pl Calculate* и свяжите его с атрибутом *M\_pl*, т.е. перетащите скрипт на атрибут *M\_pl*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметром скрипта *M\_pl Calculate*, вычисляющего относительную массу планера самолета без силовой установки, является атрибут *\_N\_pas* (рис. 3).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *M\_pl 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:= 0.44 - 0.0003 * "_N_pas";
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.

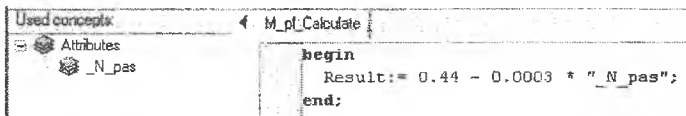


Рис. 3. Атрибуты и тело скрипта *M\_pl Calculate*

➤


### 2.2.3.3 Определение скрипта для вычисления относительной массы силовой установки

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


➤

- Создайте концепт «скрипт», который будет вычислять относительную массу силовой установки. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *M\_su Calculate* и свяжите его с атрибутом *M\_su*, т.е. перетащите скрипт на атрибут *M\_su*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметром скрипта *M\_su Calculate*, вычисляющего относительную массу силовой установки, является атрибут *\_N\_pas* (рис. 4).

➤

- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *M\_su 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:= 0.14 - 0.0002 * "_N_pas";
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.

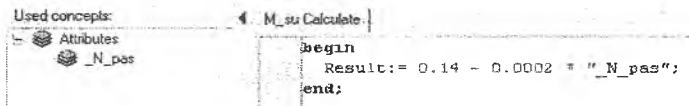


Рис. 4. Атрибуты и тело скрипта *M<sub>su</sub> Calculate*



### 2.2.3.4 Определение скрипта для вычисления взлетной массы самолета первого приближения


Чтобы вычислить с помощью скрипта взлетную массу самолета первого приближения по формуле (1), необходимо выполнить следующие действия:




➤ Создайте концепт «скрипт», который будет вычислять взлетную массу самолета первого приближения. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.

➤ Созданный скрипт переименуйте в *M<sub>0</sub> Calculate* и свяжите его с атрибутом *M<sub>0</sub>*, т.е. перетащите скрипт на атрибут *M<sub>0</sub>*.

➤ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *M<sub>0</sub> Calculate*, вычисляющего взлетную массу самолета первого приближения, являются атрибуты *M<sub>kn</sub>*, *Motn\_topl*, *M<sub>pl</sub>*, и *M<sub>su</sub>* (рис. 5).

➤ Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *M<sub>0</sub> 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:= ("M_kn"+300) / (1-"Motn_topl"- "M_pl"- "M_su");
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.

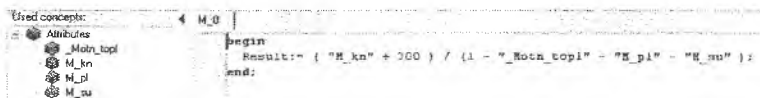




Рис. 5. Атрибуты и тело скрипта *M<sub>0</sub> Calculate*



### 2.2.3.5 Определение скрипта для вычисления минимальной потребной взлетной тяги двигателя

Чтобы вычислить с помощью скрипта минимальную потребную взлетную тягу двигателя по формуле (5), необходимо выполнить следующие действия:

- Создайте концепт «скрипт», который будет вычислять минимальную потребную взлетную тягу двигателя. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *P\_0 Calculate* и свяжите его с атрибутом *P\_0*, т.е. перетащите скрипт на атрибут *P\_0*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *P\_0 Calculate*, вычисляющего минимальную потребную взлетную тягу двигателя, являются атрибуты *P\_0*, *M\_0* и *N\_dvig* (рис. 6).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *P\_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:= "_P0" * "M_0" / "_N_dvig";
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.

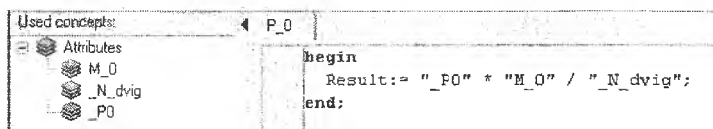



Рис. 6. Атрибуты и тело скрипта *P\_0 Calculate*


➤

### 2.2.3.6 Определение скрипта для вычисления максимального потребного удельного расхода топлива двигателя на крейсерском режиме

Чтобы вычислить с помощью скрипта максимальный потребный удельный расход топлива двигателя на крейсерском режиме по формуле (6), необходимо выполнить следующие действия:

- 
- Создайте концепт «скрипт», который будет вычислять максимальный потребный удельный расход топлива двигателя 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 *C\_kr Calculate* и свяжите его с атрибутом *C\_kr*, т.е. перетащите скрипт на атрибут *C\_kr*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипта (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *C\_kr Calculate*, вычисляющего максимальный потребный удельный расход топлива двигателя на крейсерском режиме, являются атрибуты *Motn\_topl*, *K\_max*, *V\_kr* и *L* (рис. 7).
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *C\_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 := 2.8 * "Motn_topl" * "K_max" * "V_kr" /
        ( "L" * (1-0.5*"Motn_topl") * (1.5-0.00003*"L" ) );
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.

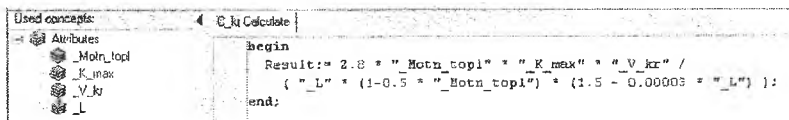



Рис. 7. Атрибуты и тело скрипта *C\_kr Calculate*

➤




### 2.2.3.7 Определение скрипта для вычисления максимальной потребной массы двигателя

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

- 
- Создайте концепт «скрипт», который будет вычислять максимальную потребную массу двигателя по формуле. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- Созданный скрипт переименуйте в *M\_dvig Calculate* и свяжите его с атрибутом *M\_dvig*, т.е. перетащите скрипт на атрибут *M\_dvig*.
- Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *M\_dvig Calculate*, вычисляющего максимальную потребную массу двигателя по формуле, являются атрибуты *M\_0*, *M\_su*, и *N\_dvig* (рис. 8).
- 
- Напишите тело скрипта (т.е. собственно скрипт). Для этого выделите скрипт *M\_dvig 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 * "M_0" * "M_su" / "N_dvig";
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.

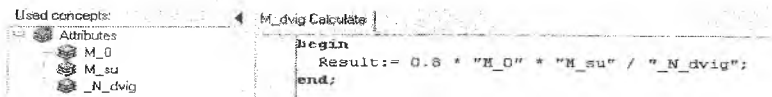


Рис. 8. Атрибуты и тело скрипта *M\_dvig Calculate*

- Таким образом, в дереве концептов дескриптивной онтологии предметной области “Выбор двигателя для проектируемого самолета” можно видеть следующие скрипты (рис. 9).

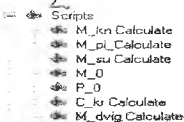


Рис. 9. Скрипты дескриптивной онтологии

## 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* -> *Arrange*.

В левой части открывшегося окна *Ontology Network* дескриптивная онтология отображается в виде дерева концептов, а в правой части – в виде семантической сети (рис. 10).

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

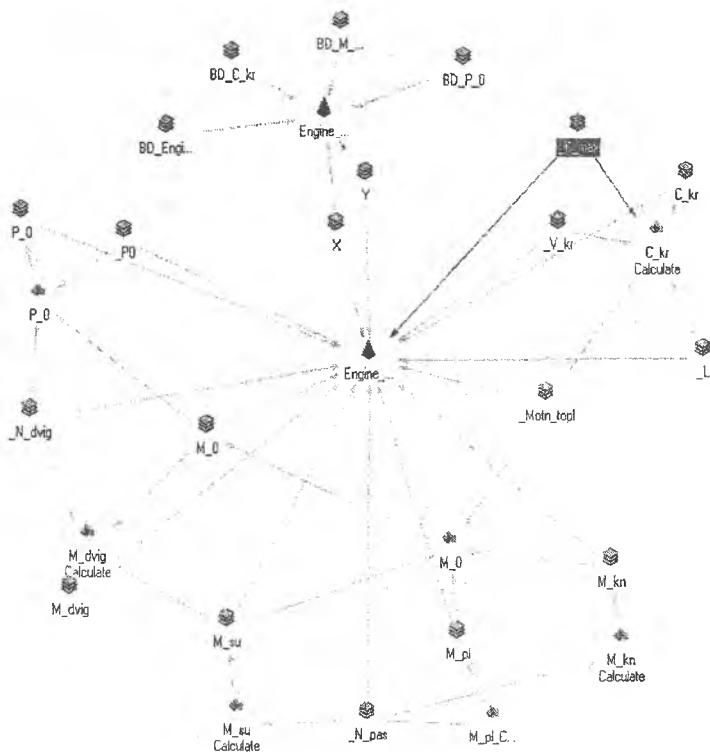


Рис. 10. Представление онтологии предметной области “Выбор двигателя для проектируемого самолета” в виде семантической сети

## 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. 11).

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. 11).

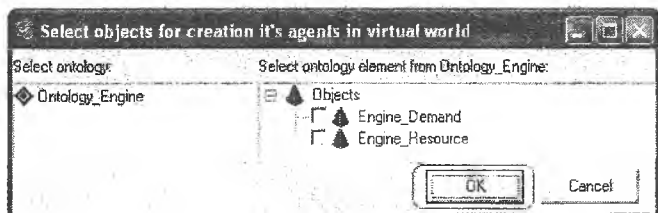


Рис. 11. Выбор объектов, для которых должны создаваться агенты

- 
- 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 Engine*. Раскройте дерево концептов онтологии виртуального мира (рис. 12).
-

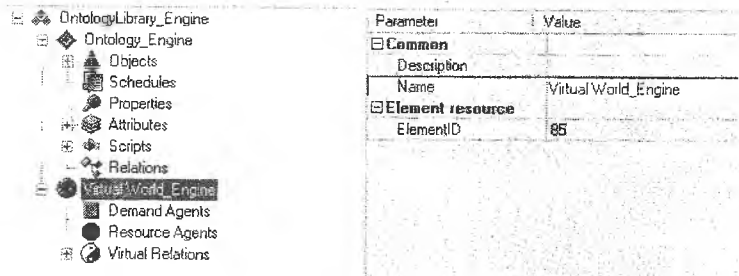


Рис. 12. Категории концептов онтологии мира заказов и ресурсов

### 2.3.2 Создание концепта «агент заказа»

- 
- Создайте концепт «агент заказа» для концепта «двигатель проектируемого самолета» (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 *Engine Demand*. Затем нажмите <OK> и переименуйте созданный концепт в *Engine\_Demand* (рис. 13).
- 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).
- 

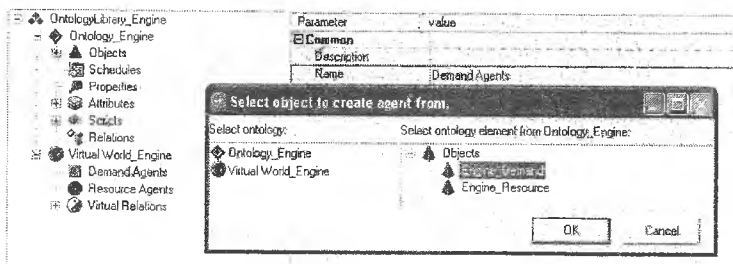


Рис. 13. Создание агента заказа для концепта *Engine\_Demand*

### 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 *Engine\_Resource*. Затем нажмите <OK> и переименуйте созданный концепт в *Engine\_Resource* (рис. 14).

- 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*.
- 

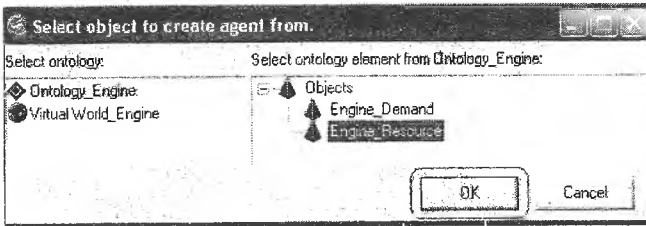


Рис. 14. Создание агента ресурса для концепта *Engine\_Resource*

### 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 *Engine\_Demand* и агента ресурса *Engine\_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 *Engine\_Demand*, а в качестве *Matching object* – *Engine\_Resource* (рис. 15).
- 

Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Engine\_Demand.Engine\_Resource* (рис. 15).

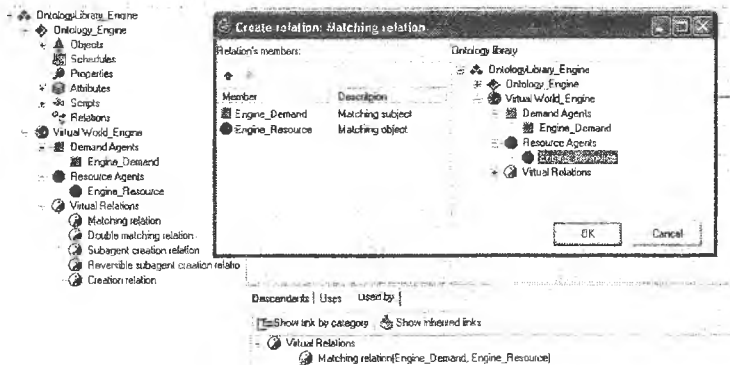


Рис. 15. Связывание отношением матчинга концептов *Engine\_Demand* и *Engine\_Resource*

### 2.3.4 Условия матчинга

- 
- Перейдите в закладку *Used by* (*Virtual relations* -> *Matching relation* -> *Used by*) и выделите *Matching relation (Engine\_Demand, Engine\_Resource)*, далее в контекстном меню выберите *Edit virtual relation properties* (рис. 16). При этом откроется окно редактирования условий матчинга (рис. 17).
- 

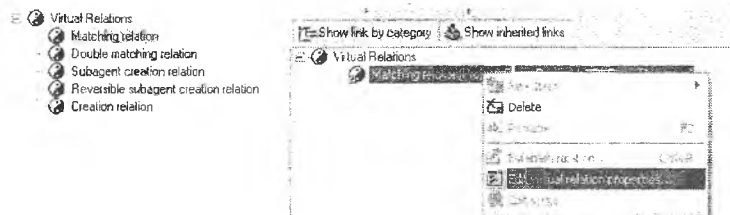


Рис. 16. Редактирование свойств отношения матчинга

В окне редактирования условий матчинга имеются следующие закладки:

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

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

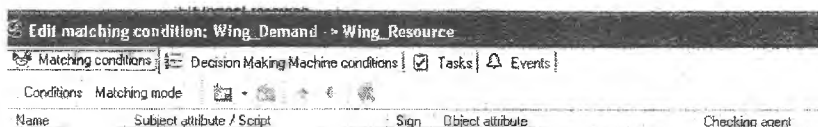




Рис. 17. Окно редактирования условий матчнга


### 2.3.4.1 Создание условий матчнга, ограничивающих выбор в базе данных двигателя для проектируемого самолета

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

В задаче выбора двигателя для проектируемого самолета параметры выбираемого двигателя должны удовлетворять следующим условиям:

- $P_0 \leq P_{0 \text{ баз}}$  – тяга двигателя, имеющегося в базе данных, должна быть больше или равна расчетной потребной взлетной тяге двигателя;
- $C_{\text{уд кр}} \geq C_{\text{уд кр баз}}$  – удельный расход топлива двигателя, имеющегося в базе данных, должен быть меньше или равен расчетному потребному удельному расходу топлива двигателя на крейсерском режиме;
- $m_{\text{дв}} \geq m_{\text{дв баз}}$  – масса двигателя, имеющегося в базе данных, должна быть меньше или равна расчетной потребной массе двигателя.



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



- 
- Создайте условие выбора двигателя по расчетной потребной взлетной тяге. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчнга. Укажите следующие параметры условия:

- SubjectAttribute = 'Engine\_Demand.P\_0';
- ObjectAttribute = ' Engine\_Resource.BD\_P\_0';
- Sign = '<='.

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

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

- Активируйте условие матчинга, выставив флажок *Active*.
- Установите свойство обязательности условия (флажок *Obligatory*).
- 
- Создайте условие выбора двигателя по расчетному потребному удельному расходу топлива двигателя на крейсерском режиме. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
  - SubjectAttribute = 'Engine\_Demand.C\_kr';
  - ObjectAttribute=' Engine\_Resource.BD\_C\_kr;
  - Sign = '>='.
- Активируйте условие матчинга, выставив флажок *Active*.
- Установите свойство обязательности условия (флажок *Obligatory*).
- 
- Создайте условие выбора двигателя по расчетной потребной массе. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
  - SubjectAttribute = 'Engine\_Demand.M\_dvig';
  - ObjectAttribute=' Engine\_Resource.BD\_M\_dvig;
  - Sign = '>='.
- Активируйте условие матчинга, выставив флажок *Active*.
- Установите свойство обязательности условия (флажок *Obligatory*).
- 

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

Условия матчинга для задачи поиска прототипа по дальности полета и количеству пассажиров приведены на рис. 18.

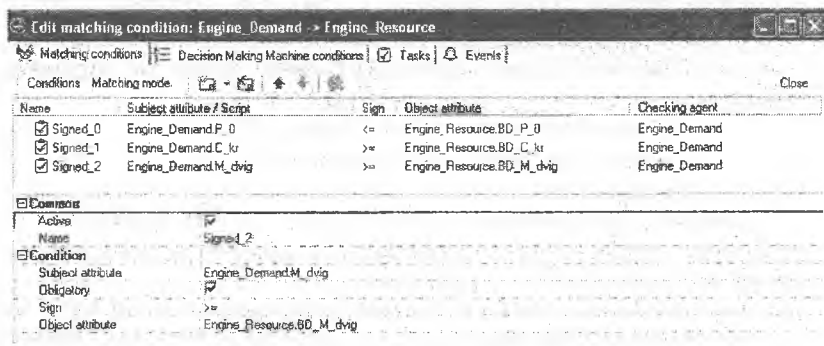




Рис. 18. Условия матчинга в задаче выбора двигателя для проектируемого самолета




### 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 – минимизация значения удельного расхода топлива на крейсерском режиме

- 
- Создайте условие 1 принятия решений для матчинга *Engine\_Demand* -> *Engine\_Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия (рис. 19):
  - Attribute = 'Engine\_Resouce.\_BD\_C\_kr';
  - Order = 'Min';
  - Weight = '100'.
- Активизируйте условие принятия решения, выставляя флажок *Active*.
- 

#### 2.3.5.2 Создание условия принятия решения – минимизация массы двигателя

- 
- Создайте условие 2 принятия решений для матчинга *Engine\_Demand* -> *Engine\_Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия (рис. 19):
  - Attribute = 'Engine\_Resouce.\_BD\_M\_dvig';
  - Order = 'Min';
  - Weight = '100'.
- Активизируйте условие принятия решения, выставляя флажок *Active*.
- 

Условия принятия решения в матчинге для выполнения расчетов приведены на рис. 19.

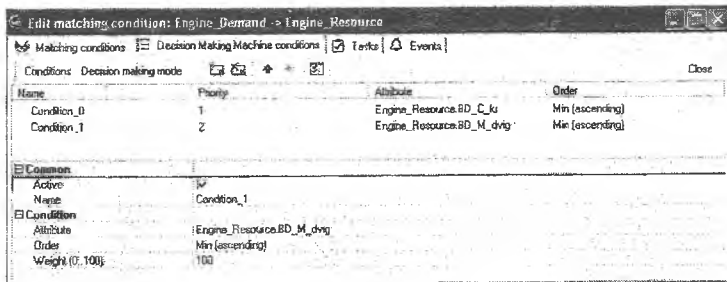


Рис. 19. Условия принятия решения в матчинге *Engine\_Demand* – *Engine\_Resource*

### 2.3.6 Параметры представления экземпляра объекта в сцене

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

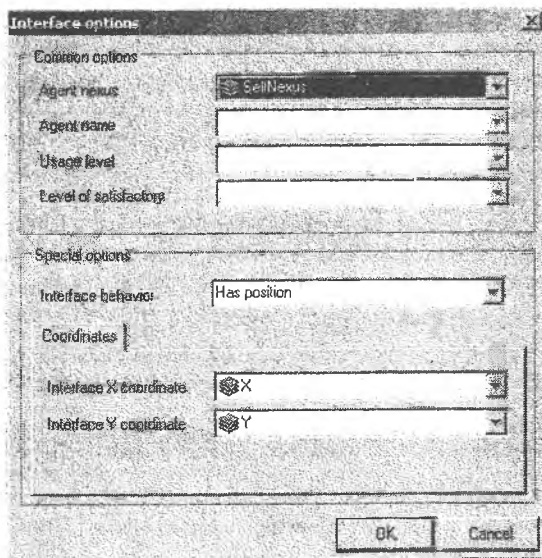


Рис. 20. Редактор свойств *Interface behaviour*

- 
- Свяжите атрибуты *X*, *Y* с позицией агента объекта *Engine\_Demand* в сцене. Для этого в редакторе свойств объекта *Engine\_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 *Engine\_Resource* в сцене. Для этого в редакторе свойств объекта *Engine\_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 двигателя для проектируемого самолета”

Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Выбор двигателя для проектируемого самолета” приведено на рис. 21.

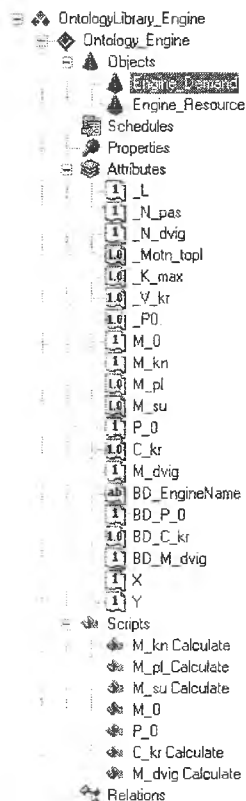



Рис. 21. Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области «Выбор двигателя для проектируемого самолета»

- 
- Сохраните созданные онтологии (дескриптивную онтологию и онтологию мира заказов/ресурсов) при помощи кнопки  под именем *Engine 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. 22).

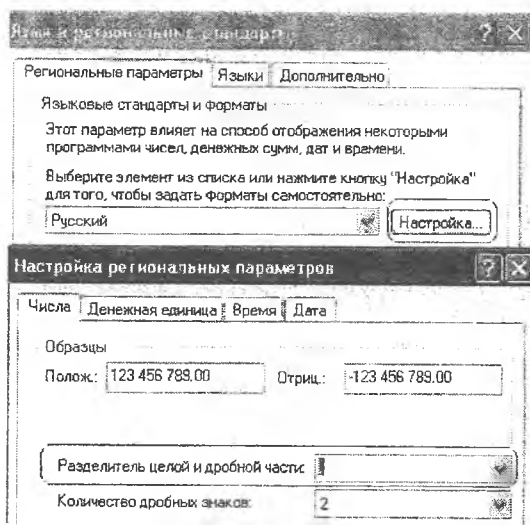



Рис. 22. Настройки компьютера

- Вызовите исполняющую систему, запуская программу  *Unintf.exe*, находящуюся в папке *OntConsUnintf*.
- Создайте новую онтологическую сцену (*File -> New scene -> Load ontology*, выберите онтологию *Engine ontology.ocl*).
- В окне физического мира создайте одного агента проектируемого самолета *Engine Demand\_1* и трех агентов двигателей-прототипов для выполнения расчетов (рис. 23). С помощью инспектора агентов установите указанные в таблице значения атрибутов для агента проектируемого самолета, например Ил-86 и для трех агентов двигателей-прототипов в базе данных.

<i>Атрибуты</i>	<i>Ил-86</i>
<i>_L</i>	4350
<i>_N_pas</i>	350
<i>_N_dvig</i>	4
<i>_Motn_topl</i>	0.3638
<i>_K_max</i>	16.47
<i>_V_kr</i>	263.6
<i>_P0</i>	0.267

<i>Имя агента</i>	<i>Тип двигателя</i>	<i>С уд кр баз</i>	<i>М дв баз, кг</i>	<i>РО баз</i>
Engine Resource 1	HK-86	0.74	2450	13000
Engine Resource 2	Д-40	0.83	2225	15315
Engine Resource 3	HK-8-2У	0.766	2170	10500

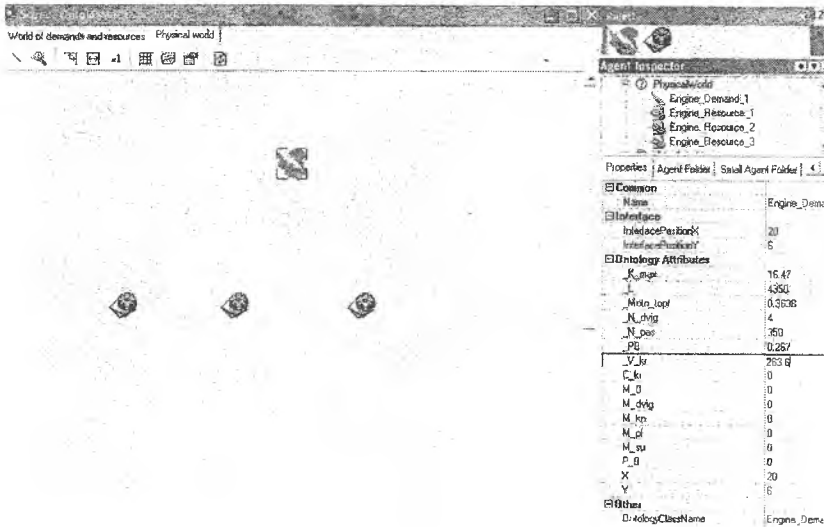



Рис. 23. Создание онтологической сцены предметной области  
«Выбор двигателя для проектируемого самолета»

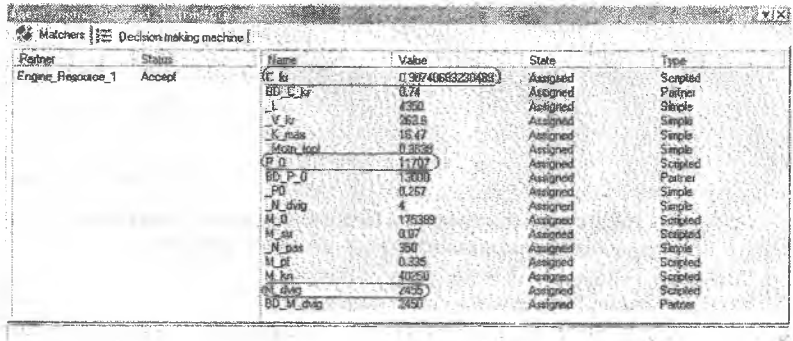
## 2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА

### 2.5.1 Запуск сцены на моделирование 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 выполнение процесса матчнинга).
- Наблюдайте процесс матчнинга между агентами проектируемого самолета и агентами самолетов-прототипов в базе данных.
- 

В процессе матчнинга активный агент проектируемого самолета (*Engine\_Demand\_1*) запускает скрипты, необходимые для вычисления значений атрибутов (рис.24). Процесс матчнинга начинается с того, что агент проектируемого самолета (*Engine\_Demand\_1*) проверяет значения атрибутов агентов двигателей-прототипов в базе данных и выбирает агентов, у которых значения атрибутов соответствуют ограничениям по тяге двигателя, удельному расходу топлива двигателя и массе двигателя.

Прототип 3 (*Engine\_Resource\_3*) не соответствует ограничению по тяге двигателя. Поэтому агент проектируемого самолета не соглашается на матчнинг с ним (синяя пунктирная стрелка в результатах матчнинга отсутствует). Далее агент проектируемого самолета строит таблицу принятия решений (рис. 26), в которой размещает двигатели-прототипы по возрастанию удельного расхода топлива, т.к. критерий минимизации удельного расхода топлива стоит первым в списке критериев принятия решений (см. 2.3.5). Агент проектируемого самолета выбирает из таблицы лучший двигатель-прототип *Engine\_Resource\_1* по критерию минимального удельного расхода топлива. Агент проектируемого самолета резервирует агента двигателя-прототипа *Engine\_Resource\_1*, который удовлетворяет его требованиям: *Engine\_Demand\_1* – *Engine\_Resource\_1* (сплошная малиновая линия в результатах матчнинга).



Partner	Status	Name	Value	State	Type
Engine_Resource_1	Accept	C_fr	1199740688320489	Assigned	Scripted
		BD_C_fr	0.74	Assigned	Partner
		L	4350	Assigned	Simple
		V_fr	260.8	Assigned	Simple
		K_mas	15.47	Assigned	Simple
		Mean_topl	0.2538	Assigned	Simple
		P_D	11707	Assigned	Scripted
		BD_P_D	1.600	Assigned	Partner
		P_D	0.257	Assigned	Simple
		N_dvlg	4	Assigned	Simple
		M_D	176389	Assigned	Scripted
		M_st	0.07	Assigned	Scripted
		N_pos	350	Assigned	Simple
		M_of	0.235	Assigned	Scripted
		M_kn	40250	Assigned	Scripted
		M_dvlg	2425	Assigned	Scripted
		BD_M_dvlg	2430	Assigned	Partner

Рис. 24. Структура агента проекта *Engine\_Demand\_1* самолета Ил-86

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента проектируемого самолета (рис. 25), в которой перечислены все агенты двигателей-прототипов, с которыми возможен матчнинг в соответствии с критериями принятия решения.

Agent Engine_Demand_1 structure		
Matches Decision making machine		
Agent	BD_C kr	BD_M dvkg
Engine_Resource_1	0.74	2450
Engine_Resource_2	0.83	2225

Рис. 25. Таблица принятия решений агента *Engine\_Demand\_1* самолета Ил-86

Аналогично можно посмотреть структуру агента-ресурса (рис. 26). В правой части таблицы указаны атрибуты агента-ресурса (*Simple*), которые использовались в процессе принятия решения о резервировании.

Agent Engine_Resource_1 structure					
Matches					
Partner	Status	Name	Value	State	Type
Engine_Demand_1	Passive	BD_C kr	0.74	Assigned	Simple
		BD_P 0	13002	Assigned	Simple
		BD_M dvkg	2450	Assigned	Simple

Рис. 26. Структура агента ресурса *Engine\_Resource\_1*

Окончательные результаты матчнинга можно видеть на рис. 27. В результате матчнинга была выполнена следующая операция резервирования:

- *Engine\_Demand\_1 – Engine\_Resource\_1*.

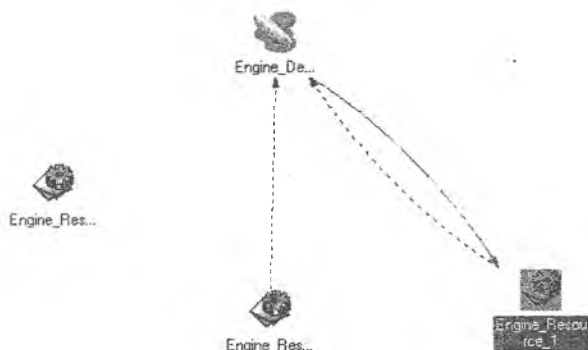



Рис. 27. Результаты матчнинга

### 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\_Engine Ил-86*. Расширение *.osf* будет добавлено автоматически. По умолчанию, сцена будет размещена в разделе *Ontology Samples*.
- Завершите работу с исполняющей системой (*File* -> *Close*).
- 

### 2.5.3. Загрузка и модификация ранее созданных сцен виртуального мира

- 
- При проведении расчетов по *Ty-154* необходимо пересчитать значения относительной массы топлива  $\bar{m}_T$  и максимального аэродинамического качества  $K_{max}$  для значения дальности полета  $L_{пол} = 3500$  км (см. лабораторную работу №2), а также значение тяговооруженности самолета  $P_0$  (см. лабораторную работу №3).
- Загрузите онтологическую сцену *Scene\_Wing Load Ty-154*: *Tools* – > *Load ontology scene* – > *Scene\_Wing Load Ty-154*. Измените в сцене значение атрибута *\_L* с 5200 на 3500.
- Запустите сцену на моделирование, получите новые значения атрибутов *Motn\_top1* и *K\_max*.
- Сохраните сцену *Tools* – > *Save ontology scene* – > *Scene\_Wing Load Ty-154*.
- 
- Загрузите онтологическую сцену *Scene\_Draught Ty-154*: *Tools* – > *Load ontology scene* – > *Scene\_Draught Ty-154*. Измените в сцене значение атрибута *\_K\_max*.
- Запустите сцену на моделирование, получите новое значение атрибута *P0\_max*.
- Сохраните сцену *Tools* – > *Save ontology scene* – > *Scene\_Draught Ty-154*.
- 

## 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 установит или изменить значения атрибутов агентов с помощью Инспектора агентов?
12. Какие действия необходимо выполнить в процессе моделирования онтологической сцены?

#### 4 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Каждый студент получает индивидуальное задание на расчет в виде ТЗ:

- дальность полета самолета, км (1000 – 10000),
- число пассажиров, чел (40 – 400),
- число двигателей на самолете (2-8) студент может выбрать самостоятельно.

Используя готовые онтологические сцены, построенные в лабораторных работах №2 и №3, и изменяя значения атрибутов, студент выполняет расчет потребных значений параметров двигателя.

На основании полученных расчетных значений и условий выбора двигателя для проектируемого самолета студент выполняет поиск соответствующего двигателя в базе данных. В условиях принятия решения можно учитывать, какое из условий является предпочтительным: минимизация удельного расхода топлива двигателя или минимизация массы двигателя.

Студент самостоятельно осуществляет заполнение онтологической сцены, используя прилагаемые таблицы данных отечественных ТРД и ТРДД, при этом ему необходимо найти данные дополнительно не менее 5-и зарубежных ТРДД, которые также должны использоваться в сцене.

Параметр									
$P_{0 \text{ баз}}$									
$C_{\text{уд кр баз}}$									
$m_{\text{дв баз}}$									

Название ТРД	P <sub>0</sub> , кгс	C <sub>уд.чр</sub> кг/кгс.ч	Мп	п <sub>др</sub> , кг
РД36-51	21000	1.23		4200
РД36-51В	7000	1.35		3860
РД36-51А	20000	1.23		4125
ТР-1	1400	1.29	0.8	865
РД-45Ф	2270	1.05	0.8	814
РД-45ФА	2270	1.05	0.8	825
0125	3000	1.055	0.8	1330

Название ТРДД	P <sub>0</sub> , кгс	C <sub>уд.чр</sub> кг/кгс.ч	Н, крейс	Мп	п <sub>др</sub> , кг
Д-20П	5500	0.88	11000	0.8	1468
Д-30 1 серии	6800	0.786	11000	0.8	1550
Д-30 2 серии	6800	0.786	11000	0.8	1768
Д-30 3 серии	6930	0.793	11000	0.8	1810
Д-30А	8400	0.72	11000	0.8	1700
Д-30КУ-154 1сер	10500	0.71	11000	0.68	2675
Д-30КУ-154 2сер	10500	0.71	11000	0.68	2675
Д-30КУ-154 3сер	10500	0.69	11000	0.68	2307
Д-30КУ 1сер	11000	0.7	11000	0.68	2650
Д-30КУ 2сер	11000	0.7	11000	0.68	2690
Д-30 КП 1 сер	12000	0.7	11000	0.68	2640
Д-30 КП 2 сер	12000	0.7	11000	0.68	2650
Д-40	15315	0.614	11000	0.68	2225
Д-70 (пр)	16000	0.632	11000	0.68	2970
Д-50 (пр)	22000	0.605	11000	0.68	2970
Д-21А1 (пр)	5380	0.95	11000	0.68	2100
Д-30КУ-90	10500	0.66	11000	0.68	2450
ПС-90А	16000	0.595	11000	0.68	2950
Д-100 (пр)	19000	0.544	11000	0.68	3500
Д-110 (пр)	22000	0.52	11000	0.68	3600
ПС-90А-76	14500	0.594	11000	0.68	2950
ПС-90А12 (пр)	12000	0.582	11000	0.68	2300
ПС-90А2 (пр)	16000	0.595	11000	0.68	2950
ПС-90АМ	16000	0.578	11000	0.68	2800
ПС-90-154	16000	0.504	11000	0.68	2950
НК-8	9500	0.83	11000	0.8	2500
НК-8-2	9500	0.79	11000	0.8	2150
НК-8-2У	10500	0.766	11000	0.8	2170
НК-8-3	9500	0.83	11000	0.8	2500

НК-8-4	10500	82.6	11000	0.8	2440
НК-8-5И	11000	0.78	11000	0.8	2400
НК-86	13000	0.74	11000	0.8	2750(6/реш)
НК-56	18000	0.625	11000	0.8	3340(6/реш)
НК-64	16000	0.58	11000	0.8	2850(6/реш)
НК-63 (пр)	30000	0.535	11000	0.8	5450
НК-104А (пр)	12000	0.58	11000	0.8	2450
НК-108 (пр)	18000	0.44	11000	0.8	3600
НК-114	14000	0.525	11000	0.8	2850
АИ-22 (пр)	3755	0.63	12000	0.75	765
АИ-25	1500	0.795	6000	0.48	348
АИ-25ТЛ	1720	0.815	6000	0.48	400
ДВ-2	2200	0.762	6000	0.48	475
Д-18Т	23430	0.57	11000	0.75	4100
Д-18Т 3серии	23430	0.57	11000	0.75	4100
Д-36	6500	0.65	11000	0.75	1106
Д-436Т	7500	0.63	11000	0.8	1490
Д-436Т1	7650	0.592	10670	0.75	1450
Д-436Т2	8190	0.592	10670	0.75	1450
Д-436ТП	7500	0.65	450	0.35	1450
Д-727М (пр)	11500	0.535	11000	0.8	2600
АИ-222-25	2500	0.84	10000	0.6	440
АИ-222-28	2800	0.81	10000	0.6	520
Х-27-2005А (пр)	11500	0.539	11000	0.8	2170
ВК-21	10388	0.63	11000	0.8	2350

*Учебное издание*

**ИСПОЛЬЗОВАНИЕ ОНТОЛОГИИ  
ПРИ ВЫБОРЕ ДВИГАТЕЛЯ  
ДЛЯ ПРОЕКТИРУЕМОГО САМОЛЕТА**

*Методические указания*

Составителя:

*Боргест Николай Михайлович  
Симонова Елена Витальевна*

Редактор Т.И. Кузнецова  
Компьютерная верстка 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 *144*. Арт. С-46/2008.

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

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