

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

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

MAGENTA CORPORATION LIMITED

Н.М.Боргест, Е.В. Симонова

ЛОГИСТИКА ВОЗДУШНОГО ФЛОТА

Методические указания к лабораторной работе № 5 по курсу

“ОНТОЛОГИЯ ПРОИЗВОДСТВЕННОЙ СФЕРЫ”

Самара
2008

1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ.....	3
2 ВЫБОР САМОЛЕТА СОГЛАСНО ТРЕБОВАНИЯМ ОТПРАВИТЕЛЯ ГРУЗА.....	4
2.1. Постановка задачи	4
2.2 ПРОЕКТИРОВАНИЕ ДЕСКРИПТИВНОЙ ОНТОЛОГИИ.....	7
2.2.1 Создание онтологии.....	7
2.2.2 Создание и удаление концепта.....	8
2.2.2.2 Концепт «атрибут»	9
2.2.2.3 Создание связей между концептами	10
2.2.3 Концепт «скрипт»	12
2.2.3.1 Определение скрипта для вычисления тарифа за перевозку одного килограмма груза	13
2.2.3.2 Определение скрипта для вычисления платы зарезервированному самолету.....	17
2.2.3.3 Определение скрипта для вычисления свободной грузоподъемности самолета.....	18
2.2.4 Представление онтологии в виде семантической сети	20
2.2.5 Параметры представления экземпляра объекта в сцене	21
2.3 ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ МИРА ЗАКАЗОВ И РЕСУРСОВ.....	23
2.3.1 Создание онтологии мира заказов и ресурсов.....	23
2.3.2 Создание концепта «агент заказа».....	24
2.3.3 Создание концепта «агент ресурса».....	25
2.3.4 Виртуальные отношения: отношение матчинга “Заказ – разделяемый ресурс”	26
2.3.5 Условия матчинга	27
2.3.5.1 Создание условий матчинга, ограничивающих выбор самолета для доставки груза	28
2.3.6 Условия принятия решения (<i>Decision Making Machine conditions</i>)	30
2.3.6.1 Алгоритм работы машины принятия решений	30
2.3.6.2 Создание условия принятия решения - минимизация значения тарифа за перевозку одного килограмма груза.....	33
2.3.6.3 Создание условия принятия решения - минимизация даты и времени вылета самолета	33
2.3.7 Матчер – сервисный объект для реализации протокола переговоров агентов в процессе матчинга	34
2.3.7.1 Задание матчеру на расчет атрибутов.....	35
2.3.7.2 Обработчики событий.....	36
2.3.7.3 Сохранение онтологий предметной области “Логистика воздушного флота”	38
2.4 СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ СЦЕНЫ	39
2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА	42
2.5.1 Запуск сцены на моделирование	42
2.5.2 Отчет по результатам моделирования сцены	48
2.5.3 Сохранение сцены виртуального мира	50
3 КОНТРОЛЬНЫЕ ВОПРОСЫ	50
4 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ.....	51

1 ЦЕЛИ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

Необходимо обеспечить перевозку грузов различных отправителей через терминал аэропорта Курумоч (Самара) с помощью самолетов различных авиакомпаний.

Основные характеристики грузов, а также требования к самолету-грузоперевозчику могут быть описаны в виде *онтологии*. При этом груз следует рассматривать как *проект* или *заказ*, а самолеты различных авиакомпаний, 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 самолетов, чьи характеристики отвечают требованиям отправителя, груз которого необходимо перевезти.

Существуют заказы, которые в процессе своего выполнения могут совместно использовать некоторый ресурс. Такой *ресурс* называется *разделяемым*. При этом ресурс может одновременно поддерживать несколько заказов, а каждый заказ должен использовать единственный ресурс. Это означает, что заказ и ресурсы находятся в отношении 1:М.

Цель лабораторной работы № 5 – разработка мультиагентных приложений, реализующих механизм принятия решений в открытых мультиагентных системах с разделяемыми ресурсами.

В процессе выполнения лабораторной работы № 5 решаются следующие задачи:

- Освоение инструментов, предоставляемых конструктором онтологий для проектирования разделяемых ресурсов.
- Освоение приемов проектирования разделяемых ресурсов, а также определения их свойств в онтологии мира заказов/ресурсов.
- Освоение приемов конструирования и моделирования онтологической сцены с использованием разделяемых ресурсов.
- Изучение матчинга между агентами заказов/ресурсов и процесса принятия решения агентом заказа по резервированию агента разделяемого 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 САМОЛЕТА СОГЛАСНО ТРЕБОВАНИЯМ ОТПРАВИТЕЛЯ ГРУЗА

2.1. Постановка задачи

На основании исходных данных по направлению перевозок, значениям предельных масс перевозимых грузов на каждом борту, потребным значениям масс перевозимых грузов от заказчиков (отправителей), необходимо выбрать в базе данных соответствующий самолет.

Рассматриваются три направления:

№	Направления
1	Самара – Москва
2	Самара – Санкт-Петербург
3	Самара – Новосибирск

Эти направления обслуживают самолеты шести авиакомпаний:

№	Авиакомпания	Количество применяемых самолетов
1	SU - Аэрофлот	Ту-134 (2 борта)
2	UT- Ютэйр	Ту-134 (1 борт), Ту-154 (2 борта)
3	E5 – Единая транспортная компания	Ту-134 (4 борта), Ту-154 (1 борт), B737 (3 борта)
4	U6 – Уральские авиалинии	Ан-24 (1 борт)
5	PL – Пулковские авиалинии	Ту-134 (2 борта)
6	X8 - Дальавиа	Ту-154 (1 борт)

Предельные массы перевозимых грузов по типам самолетов:

Марки самолетов	Предельные массы перевозимых грузов, кг
Ту-134	3000
Ту-154	6000
Ан-26	2000
B737	8000

Реальные значения предельных масс перевозимых грузов определяются за день до вылета.

Усредненное ежедневное расписание самолетов по направлениям и по компаниям (максимально возможное количество самолетовылетов в день):

Направления	Авиакомпания	Марки самолетов	Количество
Самара – Москва	SU	Ту-134	2
	UT	Ту-134	1
		Ту-154	2
	E5	Ту-134	2

		Ту-154	1
		В737	3
Самара – Санкт-Петербург	E5	Ту-134	1
	PL	Ту-134	2
Самара – Новосибирск	E5	Ту-134	1
	U6	АН-26	1
	X8	Ту-154	1

Тарифы авиакомпаний по направлениям:

Авиа-компания	Самара – Москва	Самара – Санкт-Петербург	Самара – Новосибирск
SU	Min платеж - 6.32\$/kg до 11.93 кг 0.53\$/kg до 45 кг 0.47\$/kg до 100 кг 0.42\$/kg до 300 кг 0.37\$/kg свыше 300 кг + 0.15\$/kg топливный сбор + 0.05\$/kg сбор за безопасность	-	-
UT	15 руб/кг	-	-
E5	17 руб/кг до 50 кг 14 руб/кг свыше 50 кг	21 руб/кг до 50 кг 17 руб/кг свыше 50 кг	25 руб/кг до 50 кг 22 руб/кг свыше 50 кг
PL	-	??	-
U6	-	-	??
X8	-	-	Min груз 10 кг 26 руб/кг до 45 кг 25 руб/кг до 300 кг 23 руб/кг свыше 300 кг

Пример условного расписания вылетов на неделю по направлению Самара – Москва:

День недели	Авиакомпания	Марки самолетов	Время вылета	Предельные массы перевозимых грузов, кг
1	SU	Ту-134	8-00	1000
	E5	В737	9-00	3000
	SU	Ту-134	10-00	500

	E5	Ty-154	12-00	2000
	UT	Ty-134	16-00	800
	E5	Ty-134	18-00	500
	E5	B737	20-00	5500
2	SU	Ty-134	8-00	1000
	E5	B737	9-00	3000
	SU	Ty-134	10-00	500
	UT	Ty-134	16-00	800
	E5	Ty-134	18-00	500
	E5	B737	20-00	5500
3	SU	Ty-134	8-00	1000
	E5	B737	9-00	3000
	SU	Ty-134	10-00	500
	E5	Ty-134	18-00	500
	E5	B737	20-00	5500
4	SU	Ty-134	8-00	1000
	E5	B737	9-00	3000
	SU	Ty-134	10-00	500
	E5	Ty-134	18-00	500
	E5	B737	20-00	5500
5	SU	Ty-134	8-00	1000
	E5	B737	9-00	3000
	SU	Ty-134	10-00	500
	E5	Ty-154	12-00	2000
	UT	Ty-134	16-00	800
	E5	Ty-134	18-00	500
	E5	B737	20-00	5500
6	E5	B737	9-00	3000
	SU	Ty-134	10-00	500
	E5	Ty-134	18-00	500
7	E5	B737	9-00	3000
	E5	Ty-154	12-00	2000
	E5	B737	20-00	5500

Пример условных грузов условных отправителей по направлению Самара-Москва (склад терминала):

Отправители	Груз, кг	Время и дата отправки груза
A1	100	
A2	240	
A3	350	
A4	450	

A5	120	
A6	2300	
A7	6500	
A8	1500	
A9	4500	
A10	100	
A11	40	
A12	500	
A13	750	
A14	100	
A15	240	
A16	350	
A17	450	
A18	120	
A18	2300	
A19	6500	
A20	1500	
A20	4500	
A20	100	
A21	40	

В результате матчинга необходимо получить следующие результаты:

- отправители (заказчики) должны за минимальный срок и за минимальную плату отправить грузы,
- самолеты авиакомпаний (ресурсы) должны доставить груз по назначению (до аэропорта пункта назначения). Один самолет может перевозить грузы различных отправителей.

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 *Shared resources*.
- ◆ Создайте дескриптивную онтологию предметной области «Логистика воздушного флота» (*New Item -> Descriptive ontology*). По умолчанию дескриптивная онтология имеет имя *Ontology_1*. Переименуйте ее, вводя в поле *Name* имя *Ontology_Shared resources*.

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 , 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 «Логистика воздушного флота» необходимо создать два концепта «объект»:

- *Sender* - (отправитель) с атрибутами
 - *_Sender name* (идентификатор отправителя);
 - *_Sender volume* (потребная масса груза, который необходимо перевезти);
 - *_Sender direction* (направление перевозки);
 - *_Sender date of dispatch* (время и дата отправки груза);
 - *SenderAirplane Name* (идентификатор зарезервированного самолета, на котором будет перевезен груз отправителя);
 - *Current fee* (плата отправителя зарезервированному самолету).

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

- *Airplane* - (самолет, имеющийся в базе данных) с атрибутами
 - *_Airplane name* (идентификатор самолета);
 - *_Airplane volume* (предельная масса перевозимого груза на борту, т.е. максимальная загрузка);
 - *_Airplane company* (авиакомпания);
 - *_Airplane direction* (направление);
 - *_Airplane date of dispatch* (время вылета);
 - *_Course_ \$* (курс доллара для расчета тарифа);
 - *Airplane load* (текущая загрузка самолета, кг);
 - *Airplane free volume* (свободная грузоподъемность самолета, кг);
 - *Rent fee* (тариф за перевозку 1 кг груза);
 - *Airplane account* (плата, полученная самолетом);
 - *Usage level* (загрузка самолета в процентах от максимальной

загрузки);

- *Sender names* (список идентификаторов отправителей, грузы которых размещены на самолете).

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

◆

- ◆ Создайте объект *Sender* (*Objects -> New Item -> Object*), переименуйте его в *Sender*, изменяя значение в поле *Name*, назначьте этому объекту пиктограммы, соответствующие трем видам отображения.
- ◆ Создайте объект *Airplane* (аналогично созданию *Sender*), переименуйте его в *Airplane*, 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 строковый атрибут *_Sender name* (*Attributes -> New Item -> String Attribute*), переименуйте созданный атрибут в *_Sender name*.
- ◆ Создайте вещественный атрибут *_Sender volume* (*New Item -> Float Attribute*), переименуйте созданный атрибут в *_Sender volume*.
- ◆ Создайте целый атрибут *_Sender direction* (*New Item -> Integer Attribute*), переименуйте созданный атрибут в *_Sender direction*.
- ◆ Создайте атрибут «дата-время» *_Sender date of dispatch* (*New Item -> DateTime Attribute*), переименуйте созданный атрибут в *_Sender date of dispatch*.
- ◆ Создайте строковый атрибут *SenderAirplane Name* (*New Item -> String Attribute*), переименуйте созданный атрибут в *SenderAirplane Name*.
- ◆ Создайте денежный атрибут *Currenf fee* (*New Item -> Currency Attribute*), переименуйте созданный атрибут в *Currenf fee*.

◆

- ◆ Создайте строковый атрибут *_Airplane name* (*Attributes -> New Item -> String Attribute*), переименуйте созданный атрибут в *_Airplane name*.

- ◆ Создайте вещественный атрибут *_Airplane volume* (New Item -> Float Attribute), переименуйте созданный атрибут в *_Airplane volume*.
- ◆ Создайте строковый атрибут *_Airplane company* (Attributes -> New Item -> String Attribute), переименуйте созданный атрибут в *_Airplane company*.
- ◆ Создайте строковый атрибут *_Airplane direction* (Attributes -> New Item -> String Attribute), переименуйте созданный атрибут в *_Airplane direction*.
- ◆ Создайте атрибут “дата-время” *_Airplane date of dispatch* (New Item -> DateTime Attribute), переименуйте созданный атрибут в *_Airplane date of dispatch*.
- ◆ Создайте денежный атрибут *_Course_ \$* (New Item -> Currency Attribute), переименуйте созданный атрибут в *_Course_ \$*.
- ◆ Создайте вещественный атрибут *Airplane load* (New Item -> Float Attribute), переименуйте созданный атрибут в *Airplane load*.
- ◆ Создайте вещественный атрибут *Airplane free volume* (New Item -> Float Attribute), переименуйте созданный атрибут в *Airplane free volume*.
- ◆ Создайте денежный атрибут *Rent fee* (New Item -> Currency Attribute), переименуйте созданный атрибут в *Rent fee*.
- ◆ Создайте денежный атрибут *Airplane account* (New Item -> Currency Attribute), переименуйте созданный атрибут в *Airplane account*.
- ◆ Создайте вещественный атрибут *Usage level* (New Item -> Float Attribute), переименуйте созданный атрибут в *Usage level*.
- ◆ Создайте списковый атрибут *Sender names* (New Item -> String list Attribute), переименуйте созданный атрибут в *Sender names*.
- ◆
- ◆ 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 «Логистика воздушного флота», т.е. указать, что объект *Sender* имеет атрибуты *_Sender name*, *_Sender volume*, *_Sender direction*, *_Sender date of dispatch*, *SenderAirplane Name*, *Current fee*, *X*, *Y*, а объект *Airplane* – атрибуты *_Airplane name*, *_Airplane volume*, *_Airplane company*, *_Airplane direction*, *_Airplane date of dispatch*, *_Course_ \$*, *Airplane*

load, Airplane free volume, Rent fee, Airplane account, Usage level, Sender names, X,Y, необходимо в дереве концептов «перетащить» атрибуты *_Sender name, _Sender volume, _Sender direction, _Sender date of dispatch, SenderAirplane Name, Current fee, X, Y* на объект *Sender*. Аналогично следует «перетащить» атрибуты *_Airplane name, _Airplane volume, _Airplane company, _Airplane direction, _Airplane date of dispatch, _Course_\$, Airplane load, Airplane free volume, Rent fee, Airplane account, Usage level, Sender names, X,Y* на объект *Airplane*.



◆ Установите связи между атрибутами *_Sender name, _Sender volume, _Sender direction, _Sender date of dispatch, SenderAirplane Name, Current fee, X, Y* и объектом *Sender*.

◆ Установите связи между атрибутами *_Airplane name, _Airplane volume, _Airplane company, _Airplane direction, _Airplane date of dispatch, _Course_\$, Airplane load, Airplane free volume, Rent fee, Airplane account, Usage level, Sender names, X,Y* и объектом *Airplane*.



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 *Sender* в закладке *Uses*, причем данный объект имеет атрибуты *_Sender name, _Sender volume, _Sender direction, _Sender date of dispatch, SenderAirplane Name, Current fee, X, Y* (соответствующие связи отображены в закладке *Uses*). В закладке *Used by* атрибутов отображена их связь с соответствующим объектом.

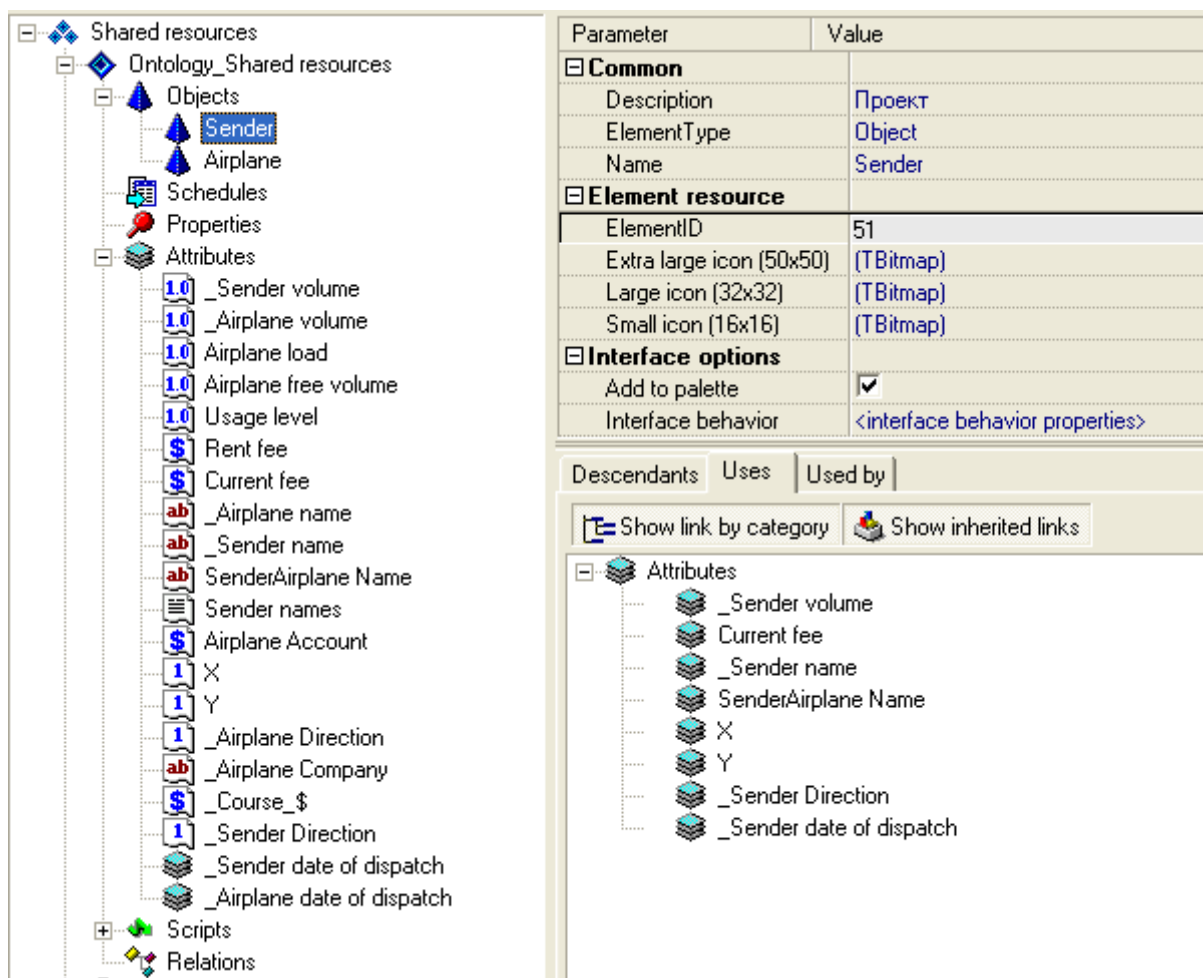



Рис. 1. Атрибуты (свойства) объекта *Sender* в закладке *Uses*

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).

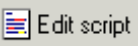
Для того чтобы указать, что некоторый концепт является параметром скрипта, необходимо «перетащить» этот концепт на соответствующий концепт «скрипт». В свою очередь, результат вычислений, производимых в скрипте, должен быть связан с каким-либо атрибутом соответствующего объекта. Для этого необходимо концепт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

Чтобы вычислить с помощью скрипта тариф за перевозку 1 килограмма груза, необходимо выполнить следующие действия:

- ◆
- ◆ Создайте концепт «скрипт», который будет вычислять тариф за перевозку 1 килограмма груза. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *Calculate Rent fee* и свяжите его с атрибутом *Rent fee*, т.е. перетащите скрипт на атрибут *Rent fee*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *Calculate Rent fee*, вычисляющего тариф за перевозку 1 килограмма груза, являются атрибуты *_Airplane direction*, *_Airplane company*, *_Course_ \$*, *_Sender volume* (рис. 2).
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Calculate Rent fee* 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
  if "_Airplane Direction" = 1 then
    begin

      if "_Airplane Company" = 'SU' then
        begin
          if ("_Sender volume"<=11.93) then
            Result:=6.32 * "_Course_ $"
          else if ("_Sender volume">11.93) and
            ("_Sender volume"<=45)
```

```

    then Result:=(0.53+0.15+0.05)* "_Course_ $"
else if ("_Sender volume">45) and
    ("_Sender volume"<=100)
    then Result:=(0.47+0.15+0.05) * "_Course_ $"
else if ("_Sender volume">100) and
    ("_Sender volume"<=300)
    then Result:=(0.42+0.15+0.05) * "_Course_ $"
else if ("_Sender volume">300)
    then Result:=(0.37+0.15+0.05) * "_Course_ $"
end

else if "_Airplane Company" = 'UT' then
begin
    if ("_Sender volume"<=5) then Result:=5*15
    else Result:=15
end

else if "_Airplane Company" = 'E5' then
begin
    if ("_Sender volume"<=25) then Result:=25*17
    else if ("_Sender volume">25) and
        ("_Sender volume"<=50) then Result:=17
    else Result:=14
end

end

else if "_Airplane Direction" = 2 then
begin

    if "_Airplane Company" = 'E5' then
begin
    if ("_Sender volume"<=25) then Result:=25*21
    else if ("_Sender volume">25) and
        ("_Sender volume"<=50) then Result:=21
    else Result:=17
end
end
end

```

```

end

else if "_Airplane Company" = 'PL' then
begin
  if ("_Sender volume"<=8.2) then Result:=396
  else if ("_Sender volume">8.2) and
    ("_Sender volume"<=45) then Result:=16.1
  else if ("_Sender volume">45) and
    ("_Sender volume"<=100) then Result:=13.2
  else if ("_Sender volume">100) and
    ("_Sender volume"<=250) then Result:=12.1
  else if ("_Sender volume">250) and
    ("_Sender volume"<=500) then Result:=11
  else if ("_Sender volume">500) then Result:=9.7
end
end

else if "_Airplane Direction" = 3 then
begin

  if "_Airplane Company" = 'E5' then
  begin
    if ("_Sender volume"<=25) then Result:=25*25
    else if ("_Sender volume">25) and
      ("_Sender volume"<=50) then Result:=25
    else Result:=22
  end

  else if "_Airplane Company" = 'X8' then
  begin
    if ("_Sender volume"<=10) then Result:=10*20
    else if ("_Sender volume">10) and
      ("_Sender volume"<=45) then Result:=20
    else if ("_Sender volume">45) and
      ("_Sender volume"<=300) then Result:=19
  end
end

```

```

else if ("_Sender volume">300) then Result:=17
end

```

```

else if "_Airplane Company" = 'U6' then
begin
  if ("_Sender volume"<=26.7) then Result:=880
  else if ("_Sender volume">26.7) and
    ("_Sender volume"<=45) then Result:=33
  else if ("_Sender volume">45) and
    ("_Sender volume"<=100) then Result:=25.3
  else if ("_Sender volume">100) then Result:=24.2
end

```

```


end

```

```

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.




```

begin
if "_Airplane Direction" = 1 then
begin
if "_Airplane Company" = 'SU' then
begin
if ("_Sender volume"<=11.93) then Result:=6.32* "_Course_$"
else if ("_Sender volume">11.93) and ("_Sender volume"<=45) then Result:=(0.53+0.15+0.0
else if ("_Sender volume">45) and ("_Sender volume"<=100) then Result:=(0.47+0.15+0.0
else if ("_Sender volume">100) and ("_Sender volume"<=300) then Result:=(0.42+0.15+0.0
else if ("_Sender volume">300) then Result:=(0.37+0.15+0.05)*"_Course_$"
end
end
else if "_Airplane Company" = 'UT' then
begin
if ("_Sender volume"<=5) then Result:=5*15
else Result:=15
end
end
else if "_Airplane Company" = 'E5' then
begin
if ("_Sender volume"<=25) then Result:=25*17
else if ("_Sender volume">25) and ("_Sender volume"<=50) then Result:=17
else Result:=14
end
end
end
else if "_Airplane Direction" = 2 then
begin

```

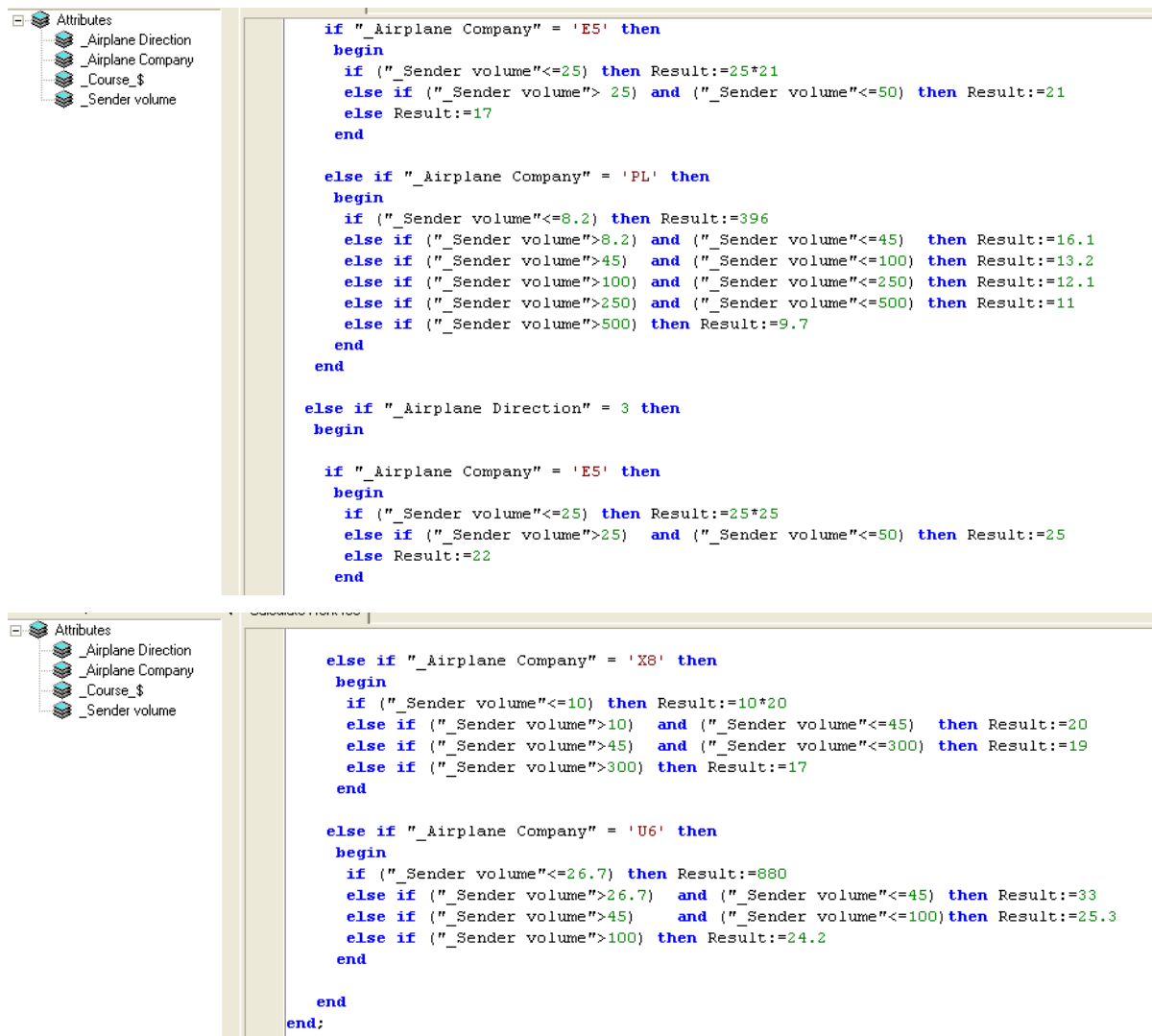



Рис. 2. Атрибуты и тело скрипта *Calculate Rent fee*




2.2.3.2 Определение скрипта для вычисления платы зарезервированному самолету

Чтобы вычислить с помощью скрипта плату зарезервированному самолету, необходимо выполнить следующие действия:



- ◆ Создайте концепт «скрипт», который будет вычислять плату зарезервированному самолету. Для этого выделите категорию *Scripts* в дереве концептов дескриптивной онтологии, затем в контекстном меню выберите *New item -> Script*.
- ◆ Созданный скрипт переименуйте в *Calculate Current fee* и свяжите его с атрибутом *Current fee*, т.е. перетащите скрипт на атрибут *Current fee*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на

концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *Calculate Current fee*, вычисляющего плату зарезервированному самолету, являются атрибуты *_Sender volume* и *Rent fee* (рис. 3).

- ◆
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Calculate Current fee* 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:= "_Sender volume" * "Rent fee";  
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 *Calculate Current fee*

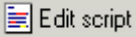
◆

2.2.3.3 Определение скрипта для вычисления свободной грузоподъемности самолета

Чтобы вычислить с помощью скрипта свободную грузоподъемность самолета, необходимо выполнить следующие действия:

◆


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

- ◆ Созданный скрипт переименуйте в *Calculate airplane free volume* и свяжите его с атрибутом *Airplane free volume*, т.е. перетащите скрипт на атрибут *Airplane free volume*.
- ◆ Укажите параметры скрипта: перетащите необходимые атрибуты на концепт скрипт (все параметры скрипта можно увидеть во вкладке *Uses*). Параметрами скрипта *Calculate airplane free volume*, свободную грузоподъемность самолета, являются атрибуты *_Airplane volume* и *Airplane load* (рис. 4).
- ◆
- ◆ Напишите тело скрипта (т.е., собственно скрипт). Для этого выделите скрипт *Calculate airplane free volume* 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 := "_Airplane volume" - "Airplane load";
```

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.

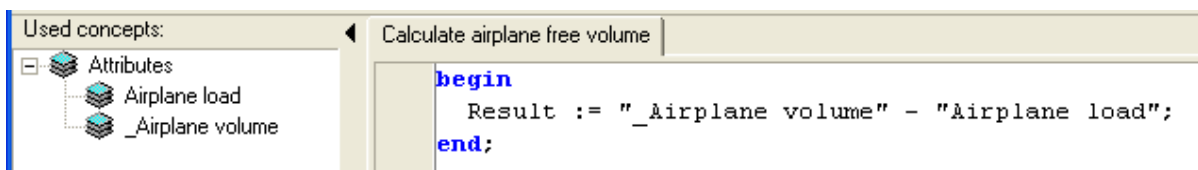


Рис. 4. Атрибуты и тело скрипта *Calculate airplane free volume*

◆

Таким образом, в дереве концептов дескриптивной онтологии предметной области “Логистика воздушного флота” можно видеть следующие скрипты (рис. 5).

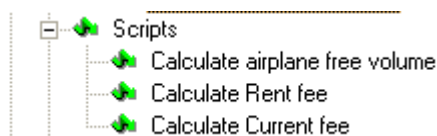


Рис. 5. Скрипты дескриптивной онтологии

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. 6).

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

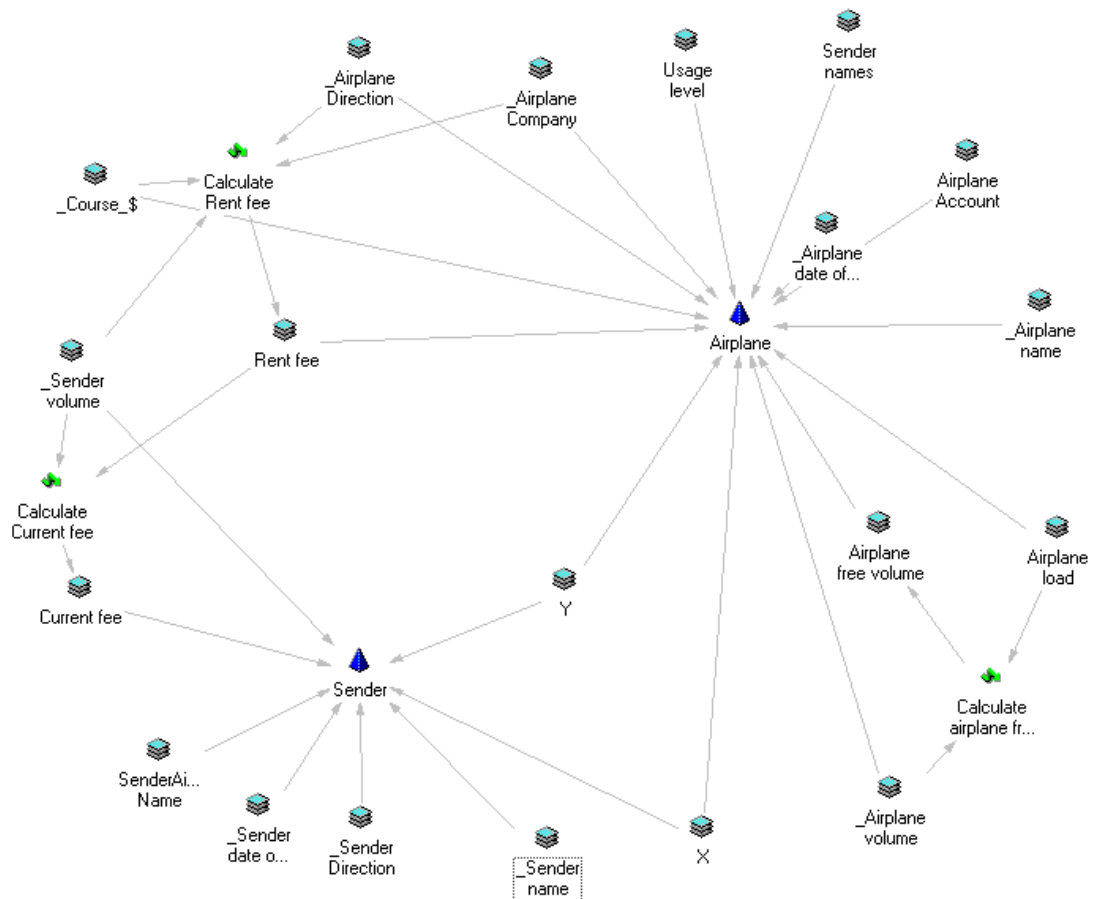


Рис. 6 Представление онтологии предметной области “Логистика воздушного флота” в виде семантической сети

2.2.5 Параметры представления экземпляра объекта в сцене

Поведение экземпляра концепта «объект» (агента) в сцене определяет Группа свойств концепта «объект» *Interface behaviour* (рис. 7).

- ◆
- ◆ Свяжите атрибуты *X*, *Y* с позицией агента объекта *Sender* в сцене. Для этого в редакторе свойств объекта *Sender_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 атрибут *_Sender name* с именем агента отправителя. Для этого в списке атрибутов *Common Options* свойства *Interface behaviour* выберите параметр *Agent name*, в появившемся диалоге выберите *_Sender name*. Нажмите <OK> (рис. 7).

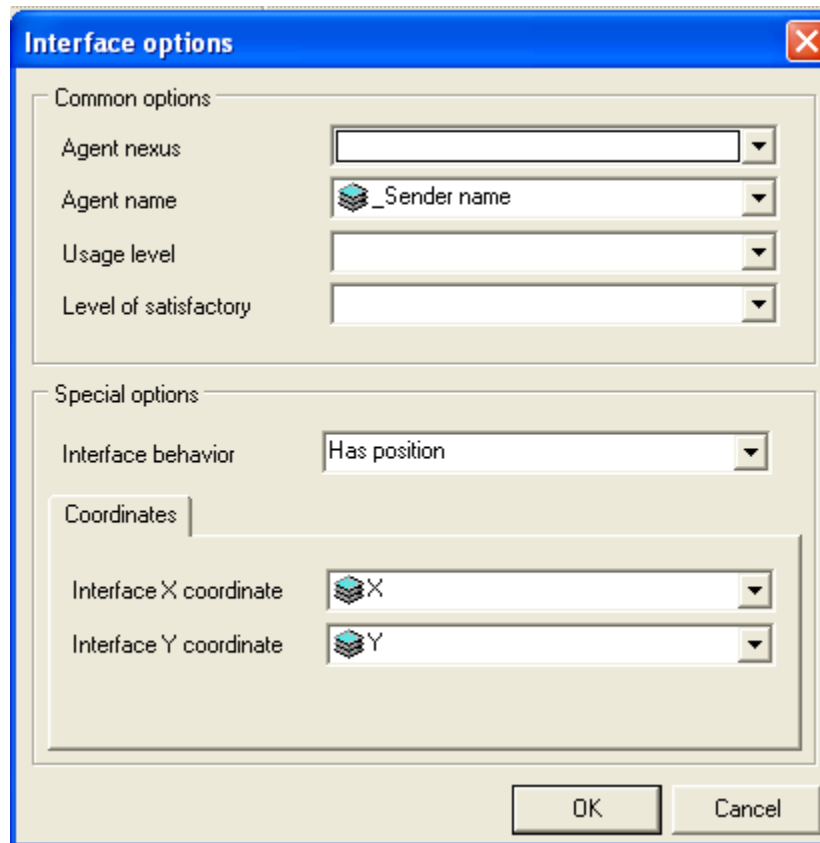


Рис. 7. Свойства агента объекта *Sender* в сцене

- ◆ Свяжите атрибуты *X*, *Y* с позицией агента объекта *Airplane* в сцене. Для этого в редакторе свойств объекта *Airplane* 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 атрибут *_Airplane name* с именем агента самолета. Для этого в списке атрибутов *Common Options* свойства *Interface behaviour* выберите параметр *Agent name*, в появившемся диалоге выберите *ServerName* (рис. 3).
- ◆ Свяжите атрибут *UsageLevel* с уровнем использования агента ресурса самолета. Для этого в списке атрибутов *Common Options* свойства *Interface behaviour* выберите параметр *Usage level*, в появившемся диалоге выберите *UsageLevel*. Нажмите <OK> (рис. 8).

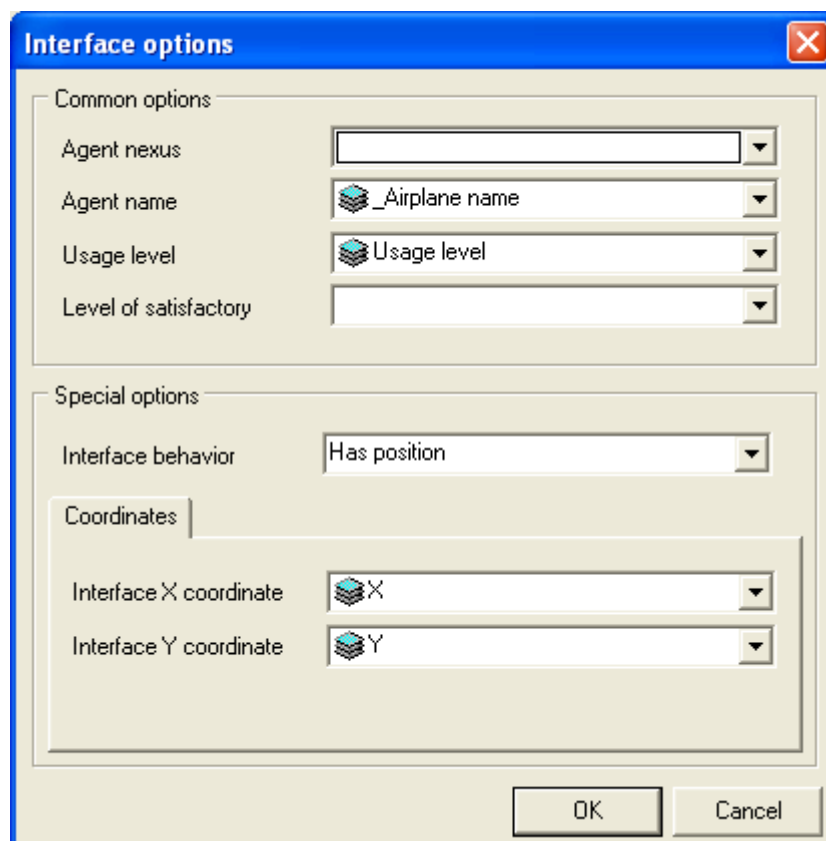


Рис. 8. Свойства агента объекта *Airplane* в сцене



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.

9).

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. 9).

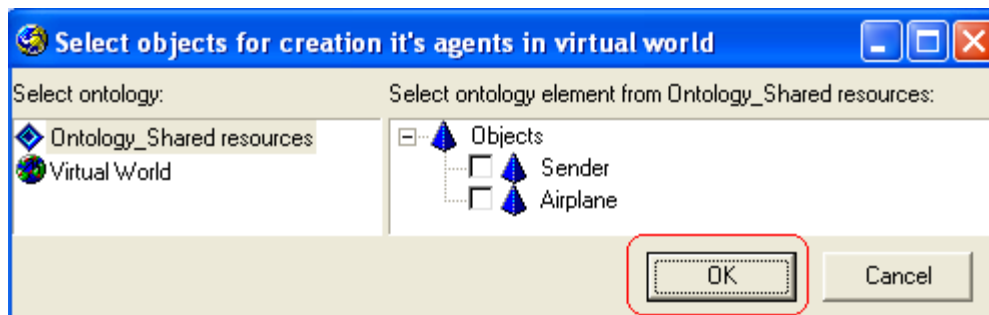


Рис. 9. Выбор объектов, для которых должны создаваться агенты

- ◆
- ◆ Создайте онтологию мира заказов и ресурсов для предметной области «Логистика воздушного флота» (*New Item -> Virtual World Ontology*). Введите в поле *Name* название онтологии - *Virtual World_Shared resources*. Раскройте дерево концептов онтологии виртуального мира (рис. 10).
- ◆

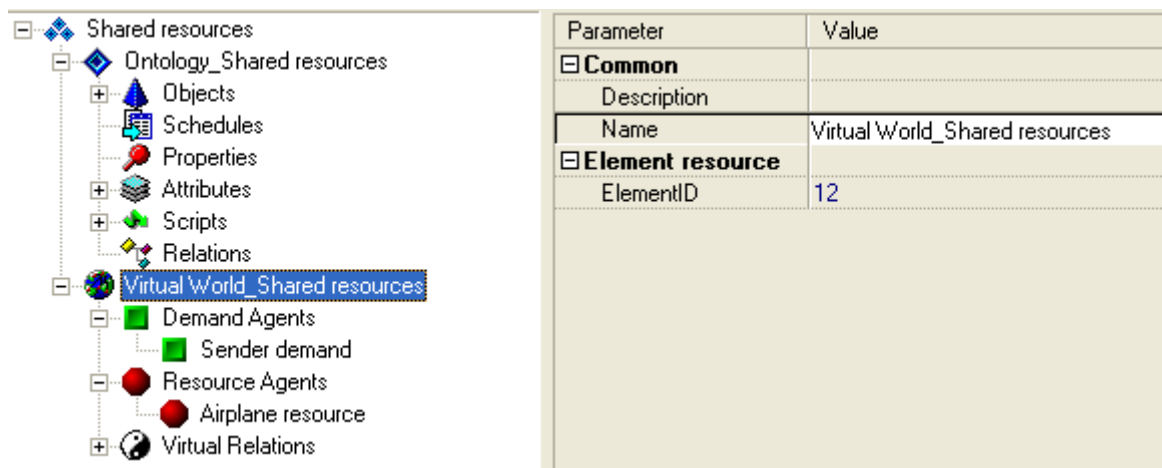


Рис. 10. Категории концептов онтологии мира заказов и ресурсов

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 *Sender*. Затем

нажмите <OK> и переименуйте созданный концепт в *Sender* (рис. 11).

- ◆ 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).
- ◆

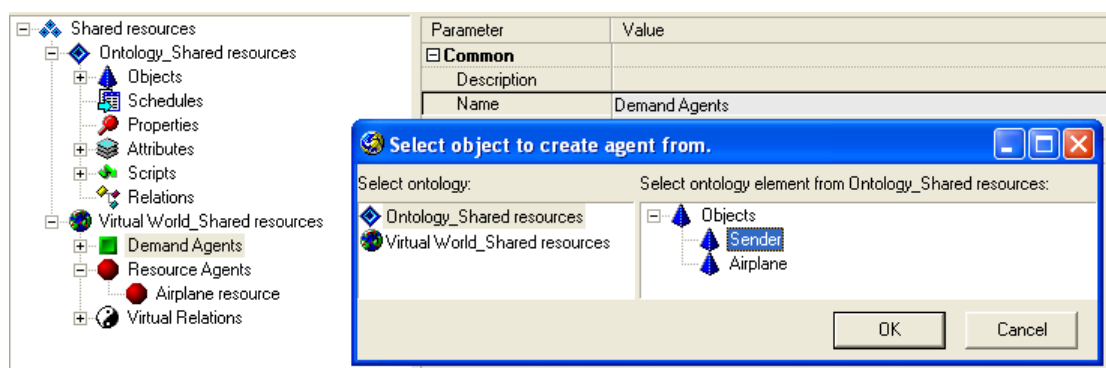


Рис. 11. Создание агента заказа для концепта *Sender*

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 *Airplane*. Затем нажмите <OK> и переименуйте созданный концепт в *Airplane* (рис. 12).
- ◆ 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*.
- ◆ Установите у концепта флаг *raoMultiUsable*, который показывает, что ресурс является **разделяемым** (рис. 13).
- ◆

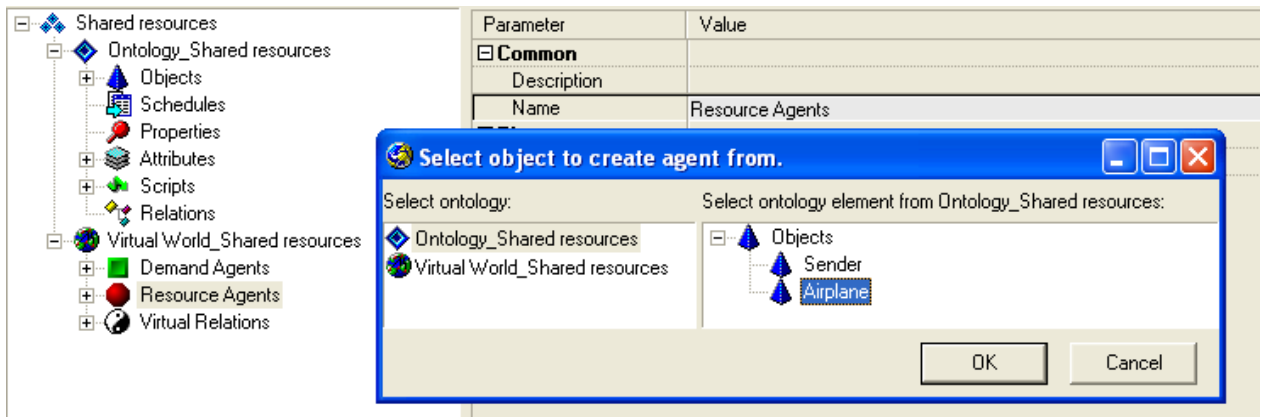


Рис. 12. Создание агента ресурса для концепта *Airplane*

Virtual world	
Options of agent	[vaoAutoCreate]
Options of resource	[raoMultiUseable]
raoActive	<input type="checkbox"/>
raoMultiUseable	<input checked="" type="checkbox"/>

Рис. 13. Установка флага *raoMultiUsable*

2.3.4 Виртуальные отношения: отношение матчинга “Заказ – разделяемый ресурс”

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 *Sender* и агента ресурса *Airplane*. 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 *Sender*, а в качестве

Matching object – Airplane (рис. 14).

Во вкладке *Used by* редактора свойств концепта «виртуальные отношения: отношение матчинга» можно видеть, что установлено отношение матчинга *Engine_Demand.Engine_Resource* (рис. 14).

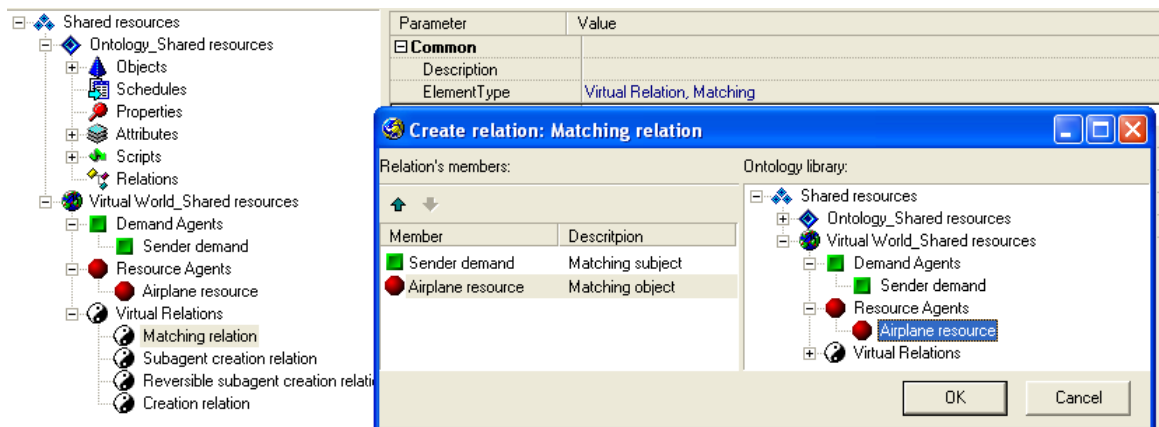


Рис. 14. Связывание отношением матчинга концептов *Sender* и *Airplane*

2.3.5 Условия матчинга

- ◆
- ◆ Перейдите в закладку *Used by* (*Virtual relations* -> *Matching relation* -> *Used by*) и выделите *Matching relation (Sender, Airplane)*, далее в контекстном меню выберите *Edit virtual relation properties* (рис. 15). При этом откроется окно редактирования условий матчинга (рис. 16).
- ◆

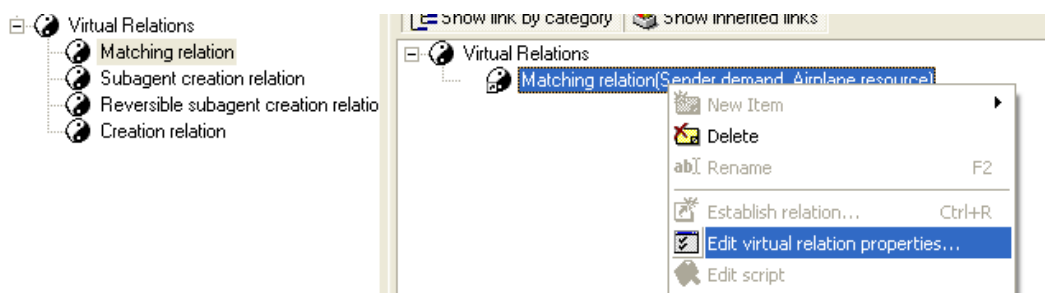


Рис. 15. Редактирование свойств отношения матчинга

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 матчере (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).

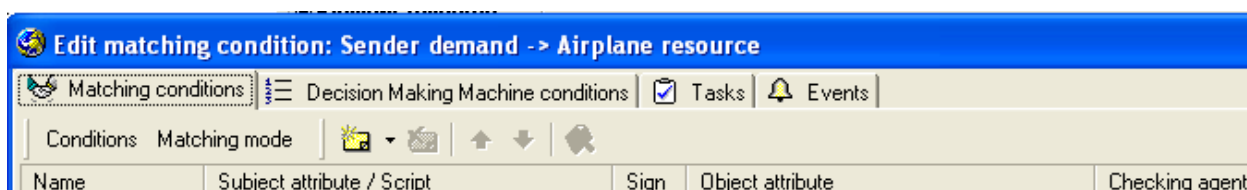




Рис. 16. Окно редактирования условий матчинга

2.3.5.1 Создание условий матчинга, ограничивающих выбор самолета для доставки груза


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

В задаче выбора самолета для доставки груза параметры выбираемого самолета должны удовлетворять следующим условиям:

- направление, обслуживаемое самолетом, должно совпадать с направлением, по которому требуется доставить груз;
- дата и время отправления груза не превышают дату и время вылета самолета согласно расписанию;
- масса перевозимого груза не должна превышать свободную часть грузоподъемности самолета, не занятую другими грузами.

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

◆



- ◆ Создайте условие выбора самолета по направлению. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:



- SubjectAttribute = 'Sender demand._Sender_Direction';
- ObjectAttribute = 'Airplane resource._Airplane_Direction';

- 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 отправителя при матчинге с агентами самолетов будет выбирать только те самолеты, направление которых совпадает с направлением, в котором должна быть осуществлена перевозка.

- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆
- ◆ Создайте условие выбора самолета по дате и времени вылета согласно расписанию. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
 - SubjectAttribute = 'Sender demand._Sender_date of dispatch';
 - ObjectAttribute = 'Airplane resource._Airplane_date of dispatch';
 - Sign = '<='.
- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆
- ◆ Создайте условие выбора двигателя по предельной массе грузов перевозимых на каждом борту. Для этого в закладке *Matching conditions* при помощи кнопки  создайте знаковое условие матчинга. Укажите следующие параметры условия:
 - SubjectAttribute = 'Sender demand._Sender_volume';
 - ObjectAttribute = 'Airplane resource.Airplane_free volume';
 - Sign = '<='.
- ◆ Активизируйте условие матчинга, выставив флажок *Active*.
- ◆ Установите свойство обязательности условия (флажок *Obligatory*).
- ◆

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

Условия матчинга для задачи поиска самолета по направлению, дате и времени вылета, а также предельной массе перевозимых грузов приведены на рис. 17.

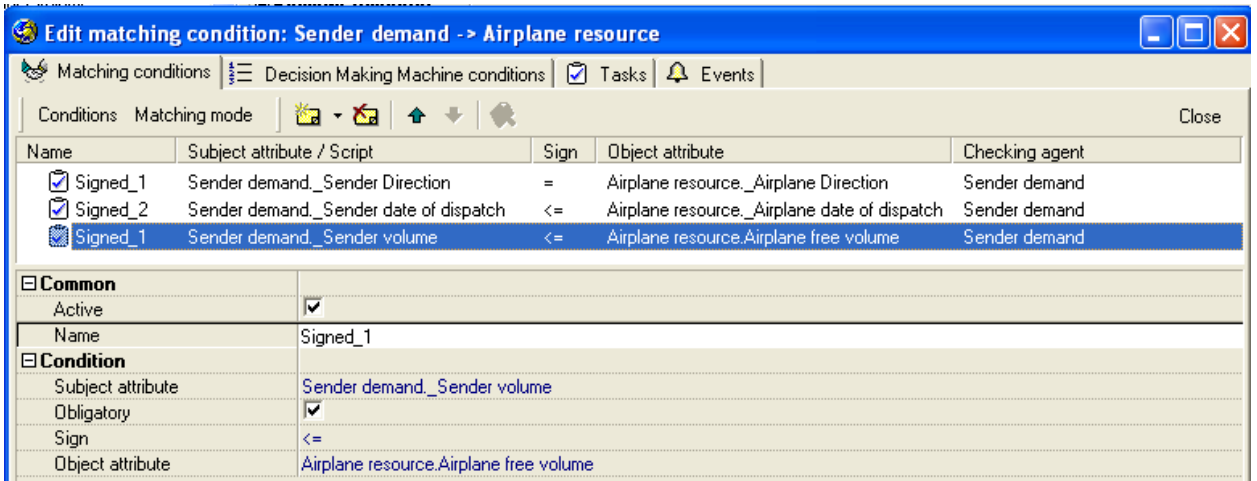




Рис. 17. Условия матчинга в задаче выбора самолета для доставки груза

2.3.6 Условия принятия решения (Decision Making Machine conditions)

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

2.3.6.1 Алгоритм работы машины принятия решений

Если задается несколько условий матчинга, дополнительно к условиям принятия решений указывается алгоритм сортировки альтернатив (метод принятия решения). Чтобы задать алгоритм сортировки альтернатив, необходимо во вкладке *Decision Making Machine conditions* нажать на кнопку  и в появившемся меню выбрать соответствующий пункт (рис. 18).

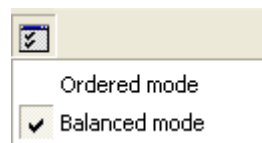


Рис. 18. Выбор метода принятия решения

Поддерживаются два алгоритма:

- *Метод главного условия принятия решений (Ordered mode)* (используется по умолчанию).

Альтернативные варианты для принятия соответствующего решения указаны в таблице X. Строки таблицы соответствуют вариантам принятия решения, а столбцы – атрибутам, на основании которых принимается

решение.

Вариант	Условие 1	Условие 2	...	Условие N
1	X_{11}	X_{12}	...	X_{1N}
2	X_{21}	X_{22}	...	X_{2N}
...
K	X_{K1}	X_{K2}	...	X_{KN}

Например, для выполнения проекта необходимо выбрать одного из трех исполнителей в соответствии с двумя критериями: максимального опыта и минимальной продолжительности выполнения проекта.

Номер альтернативы, i	Критерий, j	Опыт работы, max	Продолжительность выполнения проекта, min
	1		5
2		10	160
3		8	150
	Максимальное значение критерия	10	200

Обозначим X_{ij} – значение критерия j, соответствующее альтернативе i. Найдем максимальное значение каждого из критериев - $X_j \max$. Определим нормализованные значения критериев Y_{ij} , где

- $Y_{ij} = X_{ij} / X_j \max$, если направление оптимизации для критерия j - максимум (*soAscending*),
- $Y_{ij} = 1 - X_{ij} / X_j \max$, если направление оптимизации для критерия j - минимум (*soDescending*).

Сформируем таблицу нормализованных значений критериев Y.

Номер альтернативы, i	Нормализованный критерий, j	Опыт работы, max	Продолжительность выполнения проекта, min
	1		$5/10 = 0,5$
2		$10/10 = 1$	$1-160/200 = 0,2$
3		$8/10 = 0,8$	$1-150/200 = 0,25$
	Максимальное значение критерия	10	200

Затем альтернативы сортируются по следующему правилу. Главным считается то условие, которое указано первым в списке условий принятия решения. Условия проверяются в том порядке, как они указаны в списке условий. Если $Y_{11} > Y_{21}$, то альтернатива 1 лучше альтернативы 2; если $Y_{11} = Y_{21}$, то проверяется следующее по порядку условие ($Y_{12} > Y_{22}$) и т.д. Если первым указан критерий максимального опыта работы, то будет выбран исполнитель 2 (альтернатива 2). Если первым указан критерий минимальной продолжительности выполнения проекта, то будет выбран исполнитель 3 (альтернатива 3).

- *Метод многопараметрической средневзвешенной оптимизации (Balanced mode).*

Для условий принятия решений задаются весовые коэффициенты. Веса критериев обозначим вектором (w_1, w_2, \dots, w_n). Сформируем таблицу нормализованных значений критериев.

Номер альтернативы, i	Нормализованный критерий, j	Опыт работы, max	Продолжительность выполнения проекта, min	Обобщенная функция цели, max
1		$5/10 = 0,5$	$1-200/200 = 0$	$S_1=0,5*100+0*50 = 50$
2		$10/10 = 1$	$1-160/200 = 0,2$	$S_2=1*100+0,2*50 = 125$
3		$8/10 = 0,8$	$1-150/200 = 0,25$	$S_3=0,8*100+0,25*50 = 92,5$
Максимальное значение критерия		10	200	
Вес критерия		100	50	

Затем для каждой альтернативы (каждой i -й строки таблицы) рассчитаем обобщенную функцию цели:


$$S_i = Y_{i1} * w_1 + Y_{i2} * w_2 + \dots + Y_{in} * w_n .$$

Лучшей считается альтернатива i , для которой обобщенная функция цели принимает максимальное значение. В примере, согласно методу многопараметрической средневзвешенной оптимизации, следует выбрать альтернативу 2.

В данной задаче необходимо учесть два условия принятия решения:


- минимизация значения тарифа за перевозку одного килограмма груза;
- минимизация даты и времени вылета самолета.

Чтобы учесть баланс обоих критериев принятия решения, следует выбрать метод многопараметрической средневзвешенной оптимизации. Веса обоих критериев будем считать одинаковыми.

- ◆
- ◆ Выберите с помощью нажатия кнопки  метод принятия решений *Balanced Mode* (метод многопараметрической средневзвешенной оптимизации).


2.3.6.2 Создание условия принятия решения - минимизация значения тарифа за перевозку одного килограмма груза

Согласно данному условию, агент отправителя должен выбрать агента самолета, предоставляющего минимальный тариф за перевозку одного килограмма груза.

- ◆
- ◆ Создайте условие 1 принятия решений для матчинга *Sender_Demand* -> *Airplane_Resource*. Для этого в закладке *Decision Making Machine conditions* при помощи кнопки  создайте условие принятия решений. Укажите следующие параметры условия (рис. 19):
 - Attribute = 'Airplane resource.Rent fee';
 - Order = 'Min';
 - Weight = '100'.
- ◆ Активизируйте условие принятия решения, выставив флажок *Active*.

2.3.6.3 Создание условия принятия решения - минимизация даты и времени вылета самолета

Согласно данному условию, агент отправителя должен выбрать агента самолета, дата и время вылета которого согласно расписанию наиболее близко стоят к дате и времени отправки груза.

- ◆
- ◆ 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 = 'Airplane resource._Airplane date of dispatch';
 - Order = 'Min';

- Weight = '100'.
- ◆ Активизируйте условие принятия решения, выставив флажок *Active*.
- ◆

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

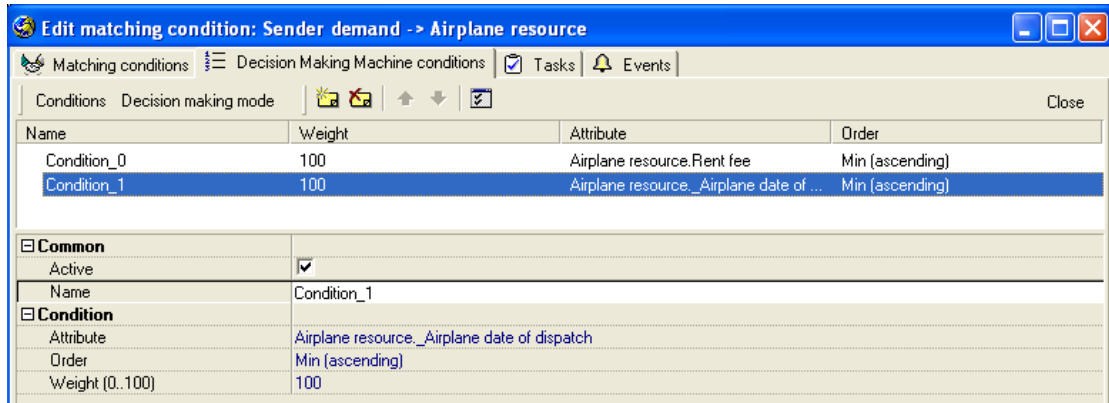


Рис. 19 – Условия принятия решения в матчинге *Sender demand* – *Airplane resource*

2.3.7 Матчер – сервисный объект для реализации протокола переговоров агентов в процессе матчинга

Объект *матчер* является сервисным объектом, который предоставляет исполняющей системе программный интерфейс для реализации протокола переговоров агентов в процессе их матчинга. Матчер выполняет следующие функции:

1. *Сбор необходимых значений атрибутов для проверки условий матчинга.*
В отношении матчинга задаются условия матчинга. В каждом из условий используются атрибуты, значения которых должны быть известны для проверки условия в конкретной ситуации. При этом сторона может не обладать всей информацией, необходимой ей для принятия решения. Например, заказу на перевозку груза необходимо знать время прибытия транспорта с грузом, чтобы сравнить с предельным сроком (deadline) и решить, подходит ему этот транспорт или нет. Поэтому матчер заказа должен «спросить» о значении этого атрибута у партнёра по матчингу, которому, в свою очередь, могут понадобиться значения атрибутов «пункт отправления», «пункт доставки» и т.д. Таким образом, матчеры «разыгрывают» сложную взаимосвязь между значениями атрибутов (простых, скриптовых, субагентов и т.д.), при этом выполняя необходимое взаимодействие с онтологией.
2. *Сбор необходимых значений атрибутов для условий выбора предложений.*
3. *Подготовка списка значений параметров для выполнения скриптов.*
Некоторые из параметров скрипта являются «простыми» атрибутами,

значения которых известны, другие параметры вычисляются с использованием скриптов. Матчер должен подготовить все необходимые значения атрибутов для вычисления скрипта, сгенерировать событие вычисления скрипта в подсистему, обслуживающую исполнение скриптов, получить результат и использовать этот результат для вычисления других скриптов, условий и т.д.



4. *Подготовка списка значений начальных условий для создания субагентов (зависимых агентов).* Субагент является дополнительным агентом, несущим информацию, необходимую для матчинга. В качестве примера можно рассмотреть задачу выбора перевозчика некоторым заказом на перевозку груза в зависимости от транспортной линии. «Главным» матчингом является матчинг между заказом на перевозку груза и перевозчиком. Для вычисления ряда атрибутов перевозчику необходимы значения таких атрибутов, как длина пути выбранного маршрута, погодные условия на маршруте и т.д. Эти атрибуты не являются атрибутами самого перевозчика, а зависят от выбранного им маршрута и вычисляются субагентом, реализующим матчинг между перевозчиком и транспортной линией.
5. *Предоставление значений для машины принятия решений.* В матчере существуют структуры для хранения полученных/вычисленных значений и существуют механизмы запроса значений по ссылке на атрибут или по имени атрибута. Такой механизм используется машиной принятия решений при заполнении таблицы значений атрибутов по вариантам матчинга.

Таким образом, матчер является неким интерфейсом между онтологией и исполняющей мультиагентной системой. С этой точки зрения, матчер представляет собой набор сервисных функций или систему, предназначенную для выполнения действий, определяемых спецификой матчинга. Матчер поддерживает взаимодействие с онтологией и агентами в сцене, а мультиагентная исполняющая система выполняет при этом свои прямые обязанности – передачу сообщений, синхронизацию и т.д., не работая напрямую с онтологией.

2.3.7.1 Задание матчеру на расчет атрибутов

Задание на определение значений атрибутов требуется в тех случаях, когда значение определенного атрибута не используется в условиях матчинга и принятия решений, но необходимо матчеру. Задание на расчет атрибута реализует механизм определения значения «дополнительного» атрибута. Дополнительные атрибуты вместе с атрибутами из условий машины принятия решений не могут быть рассчитаны ранее того момента, когда будет достигнут положительный результат матчинга (т.е. оба агента (матчера) согласятся на резервирование). Значения дополнительных атрибутов используются при написании обработчиков событий (см. пункт 2.3.7.2).

В матчинге *Sender demand* → *Airplane resource* необходимо определить задание на расчет атрибута *Current fee* (рис. 20).

- ◆
- ◆ Находясь в окне *Edit matching condition*, перейдите во вкладку *Task*, нажмите на кнопку .
- ◆ Укажите свойства задания на расчет атрибута:
- ◆ Attribute 'Sender demand.Current fee'.
- ◆ Активизируйте задание на расчет атрибута, выставив флажок *Active*.
- ◆
- ◆ Находясь в окне *Edit matching condition*, перейдите во вкладку *Task*, нажмите на кнопку .
- ◆ Укажите свойства задания на расчет атрибута:
- ◆ Attribute 'Airplane resource.Rent fee'.
- ◆ Активизируйте задание на расчет атрибута, выставив флажок *Active*.
- ◆

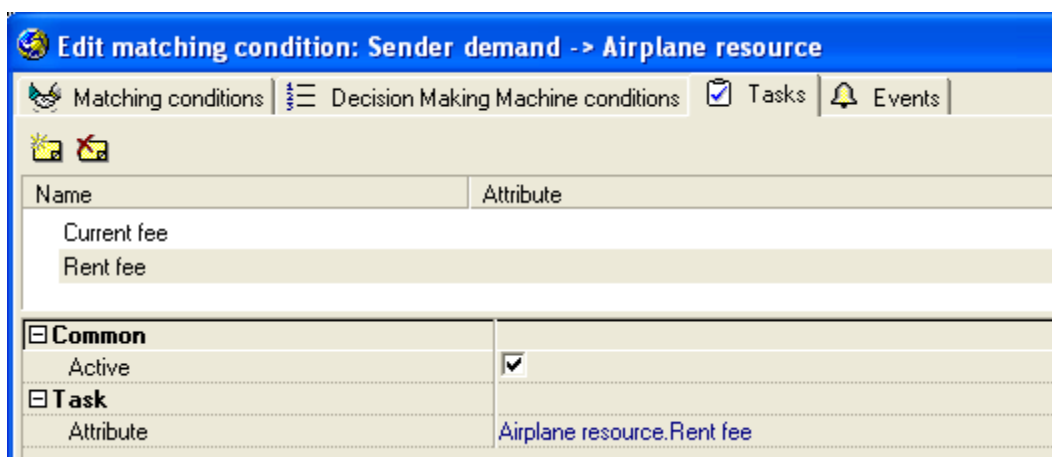


Рис. 20. Добавление задания на расчет атрибута

2.3.7.2 Обработчики событий

После того как агент заказа зарезервировал ресурса, может возникнуть необходимость в изменении значений некоторых атрибутов заказа и ресурса. Для этого используются **обработчики событий**.

Для отношения матчинга определено четыре типа событий (рис. 21):

- *Establish reservation event* - установить событие резервирования в матчинге,
- *Break reservation event* - установить событие разрыва резервирования в матчинге,

- *OnMatchingFailed* – агенты не соглашаются установить отношение матчинга,
- *ReservationFailed* – агенты устанавливают отношение матчинга, но резервирование не выполняется.

С каждым из этих событий возможно связать скрипт, который будет выполнять какие-либо действия при наступлении событий. При этом для скрипта, связанного с событием установления отношения матчинга, параметрами являются:

- *Object_Instance* – агент, который является объектом в данном отношении матчинга,
- *Subject_Instance* – агент, который является субъектом в данном отношении матчинга,
- *Matcher_Instance* – матчер агентов.

Данные события используются, когда необходимо изменить какое-то значение атрибута агента в зависимости от значения атрибутов матчера. Например, после того как отправитель зарезервировал определенную часть грузоподъемности самолета, необходимо увеличить уровень загрузки самолета на соответствующую величину, скорректировать значение занятой грузоподъемности самолета, значение платы за зарезервированную грузоподъемность и т.п.

- ◆ Находясь в окне *Edit matching condition*, перейдите во вкладку *Events*, установите флаг активности события резервирования (рис. 21).

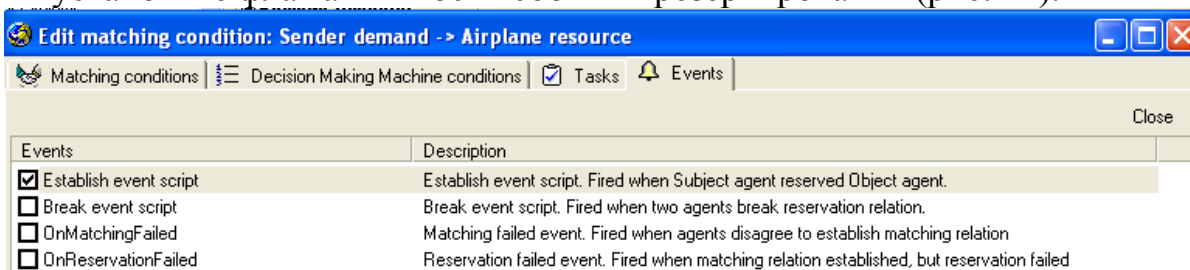



Рис. 21. Установка активности события резервирования

- ◆ Свяжите с событием резервирования скрипт. Для этого нажмите на кнопку  *Edit script...* и в появившемся редакторе скриптов наберите тело скрипта (рис. 22):

```
begin
  with "Object_Instance" do
    begin
      "Airplane load" := "Airplane load" + "Subject_Instance"."_Sender
        volume";
      "Airplane free volume" := "_Airplane volume" - "Airplane load";
      "Usage level" := 100 * "Airplane load" / "_Airplane volume";
    end
  end
end
```

```

    "Airplane account" := "Airplane account" +
    "Matcher_Instance"."Current fee";
    "Sender names".Add("Subject_Instance"."_Sender name");
    "Rent fee" := "Matcher_Instance"."Rent fee";
end;

with "Subject_Instance" do
begin
    "SenderAirplane Name" := "Object_Instance"."_Airplane name";
    "Current fee" := "Matcher_Instance"."Current fee";
end;
end;

```

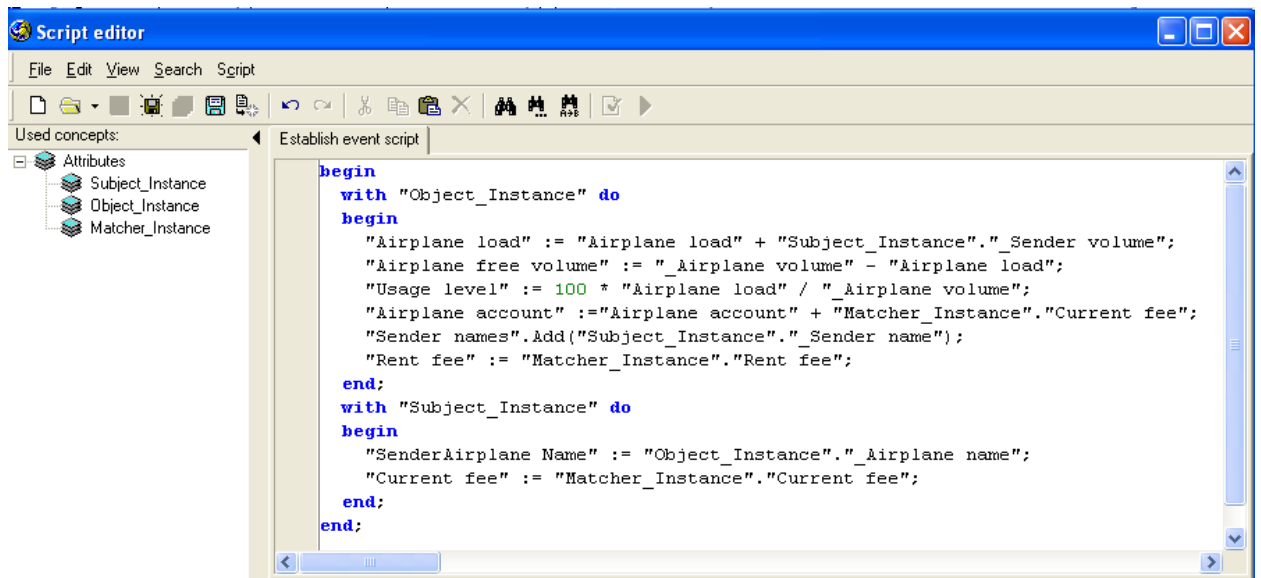


Рис. 22. Редактор скриптов обработчика событий

◆

2.3.7 Сохранение онтологий предметной области “Логистика воздушного флота”

Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Логистика воздушного флота” приведено на рис. 23.

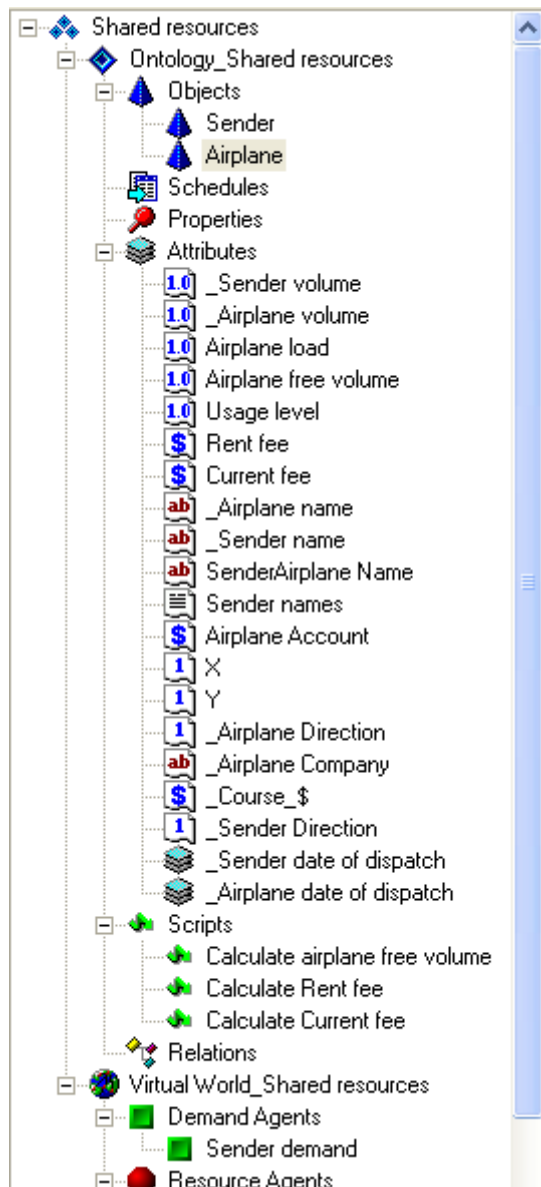



Рис. 23. Дерево концептов дескриптивной онтологии и онтологии виртуального мира предметной области “Логистика воздушного флота”

- ◆
- ◆ Сохраните созданные онтологии (дескриптивную онтологию и онтологию мира заказов/ресурсов) при помощи кнопки  под именем *Shared Airplanes 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. 24).

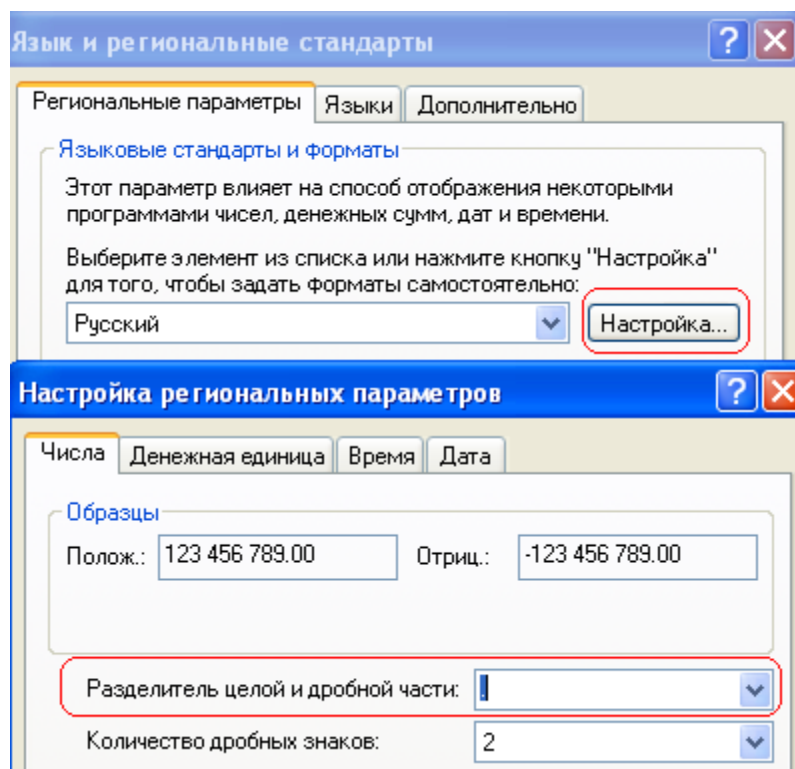



Рис. 24. Настройки компьютера

- ◆ Вызовите исполняющую систему, запуская программу  Unifntf.exe, находящуюся в папке *OntConsUniIntf*.
- ◆ Создайте новую онтологическую сцену (*File* → *New scene* → *Load ontology*), выберите онтологию *Shared resource ontology.ocl*.
- ◆ В окне физического мира создайте пятерых агентов отправителей груза и трех агентов самолетов для направления 1 (Самара-Москва) (рис. 25). С помощью инспектора агентов установите указанные в таблице значения атрибутов для агентов отправителей грузов и агентов самолетов.

Имя агента отправителя груза	Груз, кг	Дата и время отправки груза
Sender_1	100	03.04.07 6:00
Sender_2	800	03.04.07 10:00
Sender_3	700	03.04.07 16:00
Sender_4	500	03.04.07 6:45
Sender_5	700	03.04.07 6:30

Имя агента самолета	Авиакомпания	Марки самолетов	Дата и время вылета	Предельные массы, кг
Airplane_1	E5	B737	03.04.07 20:00	5500
Airplane_2	SU	Ty-134_1	03.04.07 8:00	1000
Airplane_3	UT	Ty_134_2	03.04.07 16:00	800

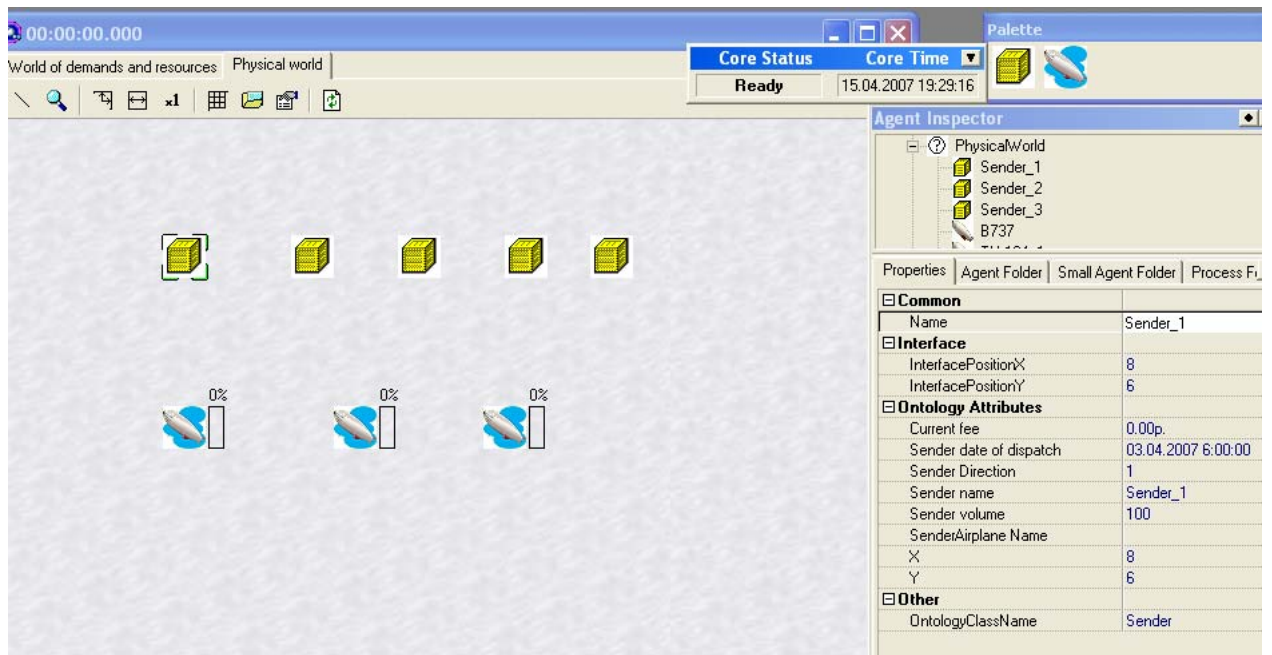


Рис. 25. Создание онтологической сцены предметной области «Логистика воздушного флота»

- ◆ Для того, чтобы установить значение атрибута типа *Date Time*, необходимо использовать специализированный редактор (рис. 26).

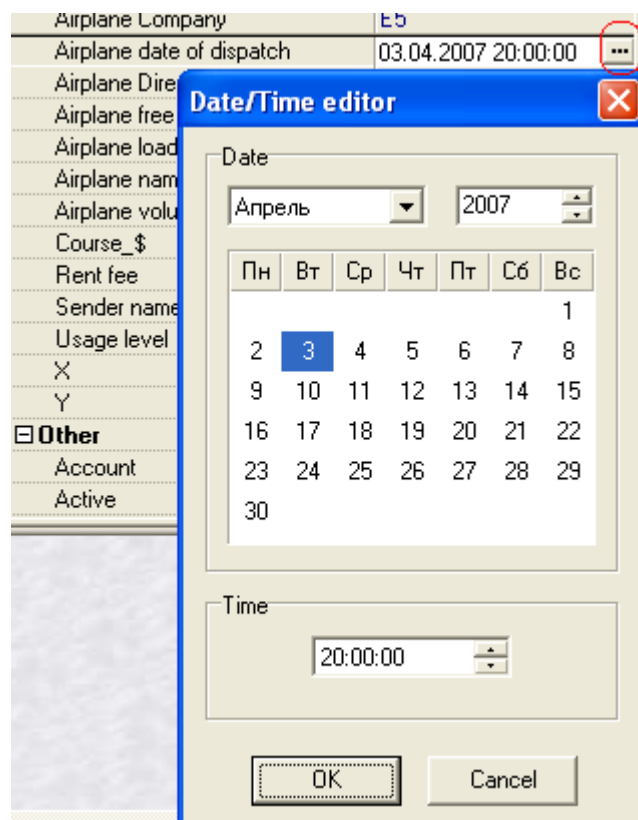



Рис. 26. Редактор значений атрибутов типа *Date Time*

◆

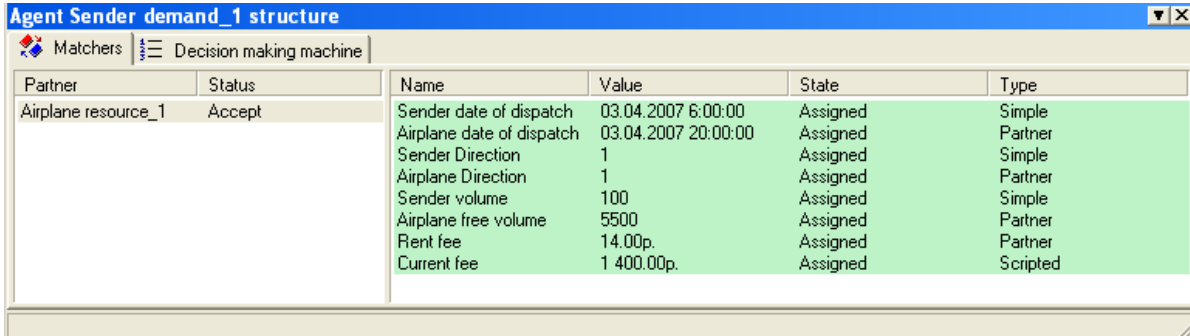
2.5 МОДЕЛИРОВАНИЕ СЦЕНЫ ВИРТУАЛЬНОГО МИРА

2.5.1 Запуск сцены на моделирование

- ◆
- ◆ 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 отправителей грузов и агентами самолетов.
- ◆

Процесс матчинга начинается с того, что агенты отправителей грузов проверяют значения атрибутов агентов самолетов и выбирают самолеты, у которых значения атрибутов соответствуют ограничениям по направлению, времени вылета и предельной массе перевозимого груза.

Агенту отправителя *Sender_1* по грузоподъемности и времени вылета подходят все самолеты (синие пунктирные стрелки в результатах матчинга). Но столь малый объем груза (100 кг) невыгодно везти на самолетах компаний SU и UT, т.к. у них более высокие тарифы, чем у E5. Поэтому, согласно методу многопараметрической средневзвешенной оптимизации, агент *Sender_1* резервирует самолет компании E5: *Sender_1 – Airplane_1* (сплошная малиновая линия в результатах матчинга). Структура агента *Sender_1* показана на рис. 27.



Partner	Status	Name	Value	State	Type
Airplane resource_1	Accept	Sender date of dispatch	03.04.2007 6:00:00	Assigned	Simple
		Airplane date of dispatch	03.04.2007 20:00:00	Assigned	Partner
		Sender Direction	1	Assigned	Simple
		Airplane Direction	1	Assigned	Partner
		Sender volume	100	Assigned	Simple
		Airplane free volume	5500	Assigned	Partner
		Rent fee	14.00p.	Assigned	Partner
		Current fee	1 400.00p.	Assigned	Scripted

Рис. 27. Структура агента *Sender_1*

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента отправителя *Sender_1* (рис. 28), в которой перечислены все агенты самолетов, с которыми возможен матчинг в соответствии с критериями принятия решения.

Agent Sender demand_1 structure		
Agent	Rent fee	Airplane date of dispatch
Airplane resource_1	14.00р.	03.04.2007 20:00:00
Airplane resource_3	15.00р.	03.04.2007 16:00:00
Airplane resource_2	16.75р.	03.04.2007 8:00:00

Рис. 28. Таблица принятия решений агента *Sender_1*

Агенту отправителя *Sender_2* по времени вылета подходят только *Airplane_1* и *Airplane_3* (синие пунктирные стрелки в результатах матчинга). Оба эти самолета подходят также и по грузоподъемности, но тариф ниже у *Airplane_1*. Кроме того, грузы отправителей *Sender_4* и *Sender_5* должны быть отправлены раньше, чем груз *Sender_2*. Если груз отправителя *Sender_2* разместить на *Airplane_3*, который улетает раньше, чем *Airplane_1*, там не смогут разместиться грузы *Sender_4* или *Sender_5*. Поэтому, согласно методу многопараметрической средневзвешенной оптимизации, агент *Sender_2* резервирует самолет компании E5: *Sender_2 – Airplane_1* (сплошная малиновая линия в результатах матчинга). Структура агента *Sender_2* показана на рис. 29.

Partner	Status	Name	Value	State	Type
Airplane resource_1	Accept	Sender date of dispatch	03.04.2007 10:00:00	Assigned	Simple
		Airplane date of dispatch	03.04.2007 20:00:00	Assigned	Partner
		Sender Direction	1	Assigned	Simple
		Airplane Direction	1	Assigned	Partner
		Sender volume	800	Assigned	Simple
		Airplane free volume	5400	Assigned	Partner
		Rent fee	14.00р.	Assigned	Partner
		Current fee	11 200.00р.	Assigned	Scripted

Рис. 29. Структура агента *Sender_2*

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента отправителя *Sender_2* (рис. 30), в которой перечислены все агенты самолетов, с которыми возможен матчинг в соответствии с критериями принятия решения.

Agent Sender demand_2 structure		
Agent	Rent fee	Airplane date of dispatch
Airplane resource_1	14.00р.	03.04.2007 20:00:00
Airplane resource_3	15.00р.	03.04.2007 16:00:00

Рис. 30. Таблица принятия решений агента *Sender_2*

Агенту отправителя *Sender_3* по времени вылета подходят только *Airplane_1* и *Airplane_3* (синие пунктирные стрелки в результатах матчинга).

Оба эти самолета подходят также и по грузоподъемности, но тариф ниже у *Airplane_1*. Кроме того, грузы отправителей *Sender_4* и *Sender_5* должны быть отправлены раньше, чем груз *Sender_3*. Если груз отправителя *Sender_3* разместить на *Airplane_3*, который улетает раньше, чем *Airplane_1*, там не смогут разместиться грузы *Sender_4* или *Sender_5*. Поэтому, согласно методу многопараметрической средневзвешенной оптимизации, агент *Sender_3* резервирует самолет компании E5: *Sender_3 – Airplane_1* (сплошная малиновая линия в результатах матчинга). Структура агента *Sender_3* показана на рис. 31.

Partner	Status	Name	Value	State	Type
Airplane resource_1	Accept	Sender date of dispatch	03.04.2007 16:00:00	Assigned	Simple
		Airplane date of dispatch	03.04.2007 20:00:00	Assigned	Partner
		Sender Direction	1	Assigned	Simple
		Airplane Direction	1	Assigned	Partner
		Sender volume	700	Assigned	Simple
		Airplane free volume	4600	Assigned	Partner
		Rent fee	14.00p.	Assigned	Partner
		Current fee	9 800.00p.	Assigned	Scripted

Рис. 31. Структура агента *Sender_3*

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента отправителя *Sender_3* (рис. 32), в которой перечислены все агенты самолетов, с которыми возможен матчинг в соответствии с критериями принятия решения.

Agent	Rent fee	Airplane date of dispatch
Airplane resource_1	14.00p.	03.04.2007 20:00:00
Airplane resource_3	15.00p.	03.04.2007 16:00:00

Рис. 32. Таблица принятия решений агента *Sender_3*

Агенту отправителя *Sender_4* по времени вылета подходят все самолеты (синие пунктирные стрелки в результатах матчинга). По грузоподъемности также подходят все самолеты, при этом самый низкий тариф у *Airplane_1*. Но, время вылета *Airplane_2* ближе к требуемому времени отправления *Sender_4*, чем *Airplane_1*. *Sender_5* должен быть отправлен раньше, чем *Sender_4*. И по времени вылета *Airplane_2* также подходит *Sender_5*. Но, масса *Sender_4* меньше, чем масса *Sender_5*, поэтому, согласно методу многопараметрической средневзвешенной оптимизации, агент *Sender_4* резервирует самолет компании SU: *Sender_4 – Airplane_2* (сплошная малиновая линия в результатах матчинга). Структура агента *Sender_4* показана на рис. 33.

Partner	Status	Name	Value	State	Type
Airplane resource_2	Accept	Sender date of dispatch	03.04.2007 6:45:00	Assigned	Simple
		Airplane date of dispatch	03.04.2007 8:00:00	Assigned	Partner
		Sender Direction	1	Assigned	Simple
		Airplane Direction	1	Assigned	Partner
		Sender volume	500	Assigned	Simple
		Airplane free volume	1000	Assigned	Partner
		Rent fee	14.25p.	Assigned	Partner
		Current fee	7 125.00p.	Assigned	Scripted

Рис. 33. Структура агента *Sender_4*

При переходе в закладку *Decision Making Machine* открывается таблица принятия решений агента отправителя *Sender_4* (рис. 34), в которой перечислены все агенты самолетов, с которыми возможен матчнинг в соответствии с критериями принятия решения.

Agent	Rent fee	Airplane date of dispatch
Airplane resource_2	14.25p.	03.04.2007 8:00:00
Airplane resource_1	14.00p.	03.04.2007 20:00:00
Airplane resource_3	15.00p.	03.04.2007 16:00:00

Рис. 34. Таблица принятия решений агента *Sender_4*

Агенту отправителя *Sender_5* по времени вылета подходят все самолеты, ближе всех по времени вылета *Airplane_2*, но оставшейся после размещения на нем *Sender_4* грузоподъемности уже не достаточно для размещения *Sender_5* (синие пунктирные стрелки в результатах матчнинга). *Airplane_1* и *Airplane_3* подходят по грузоподъемности, но, по времени вылета ближе *Airplane_3*. Несмотря на то, что тариф у *Airplane_3* выше, чем у *Airplane_1*, согласно методу многопараметрической средневзвешенной оптимизации, агент *Sender_5* резервирует самолет компании UT: *Sender_5* – *Airplane_3* (сплошная малиновая линия в результатах матчнинга). Структура агента *Sender_5* показана на рис. 35.

Partner	Status	Name	Value	State	Type
Airplane resource_3	Accept	Sender date of dispatch	03.04.2007 6:30:00	Assigned	Simple
		Airplane date of dispatch	03.04.2007 16:00:00	Assigned	Partner
		Sender Direction	1	Assigned	Simple
		Airplane Direction	1	Assigned	Partner
		Sender volume	700	Assigned	Simple
		Airplane free volume	800	Assigned	Partner
		Rent fee	15.00p.	Assigned	Partner
		Current fee	10 500.00p.	Assigned	Scripted

Рис. 35. Структура агента *Sender_5*

При переходе в закладку *Decision Making Machine* открывается таблица

принятия решений агента отправителя *Sender_5* (рис. 36), в которой перечислены все агенты самолетов, с которыми возможен матчинг в соответствии с критериями принятия решения.

Agent Sender demand_5 structure		
Agent	Rent fee	Airplane date of dispatch
Airplane resource_3	15.00p.	03.04.2007 16:00:00
Airplane resource_1	14.00p.	03.04.2007 20:00:00

Рис. 36. Таблица принятия решений агента *Sender_6*

Аналогично можно посмотреть структуру агентов-ресурсов (рис. 37 - 41). В правой части таблицы указаны атрибуты агентов-ресурсов, используемых в процессе принятия решения о резервировании, которые задавались в качестве исходных данных (*Simple*), вычислялись (*Scripted*) или были получены от агента-партнера по матчингу (*Partner*).

Agent Airplane resource_1 structure						
Partner	Status	Name	Value	State	Type	
Sender demand_1	Passive	Airplane date of dispatch	03.04.2007 20:00:00	Assigned	Simple	
Sender demand_2	Passive	Airplane Direction	1	Assigned	Simple	
Sender demand_3	Passive	Airplane free volume	5500	Assigned	Scripted	
		Airplane volume	5500	Assigned	Simple	
		Airplane load	0	Assigned	Simple	
		Rent fee	14.00p.	Assigned	Scripted	
		Sender volume	100	Assigned	Partner	
		Course_ \$	25.00p.	Assigned	Simple	
		Airplane Company	E5	Assigned	Simple	

Рис. 37. Структура агента ресурса *Airplane_1* по результатам матчинга с *Sender_1*

Agent Airplane resource_1 structure						
Partner	Status	Name	Value	State	Type	
Sender demand_1	Passive	Airplane date of dispatch	03.04.2007 20:00:00	Assigned	Simple	
Sender demand_2	Passive	Airplane Direction	1	Assigned	Simple	
Sender demand_3	Passive	Airplane free volume	5400	Assigned	Scripted	
		Airplane volume	5500	Assigned	Simple	
		Airplane load	100	Assigned	Simple	
		Rent fee	14.00p.	Assigned	Scripted	
		Sender volume	800	Assigned	Partner	
		Course_ \$	25.00p.	Assigned	Simple	
		Airplane Company	E5	Assigned	Simple	

Рис. 38. Структура агента ресурса *Airplane_1* по результатам матчинга с *Sender_2*

Agent Airplane resource_1 structure					
Matchers					
Partner	Status	Name	Value	State	Type
Sender demand_1	Passive	Airplane date of dispatch	03.04.2007 20:00:00	Assigned	Simple
Sender demand_2	Passive	Airplane Direction	1	Assigned	Simple
Sender demand_3	Passive	Airplane free volume	4600	Assigned	Scripted
		Airplane volume	5500	Assigned	Simple
		Airplane load	900	Assigned	Simple
		Rent fee	14.00p.	Assigned	Scripted
		Sender volume	700	Assigned	Partner
		Course_ \$	25.00p.	Assigned	Simple
		Airplane Company	E5	Assigned	Simple

Рис. 39. Структура агента ресурса *Airplane_1* по результатам матчинга с *Sender_3*

Agent Airplane resource_2 structure					
Matchers					
Partner	Status	Name	Value	State	Type
Sender demand_4	Passive	Airplane date of dispatch	03.04.2007 8:00:00	Assigned	Simple
		Airplane Direction	1	Assigned	Simple
		Airplane free volume	1000	Assigned	Scripted
		Airplane volume	1000	Assigned	Simple
		Airplane load	0	Assigned	Simple
		Rent fee	14.25p.	Assigned	Scripted
		Sender volume	500	Assigned	Partner
		Course_ \$	25.00p.	Assigned	Simple
		Airplane Company	SU	Assigned	Simple

Рис. 40. Структура агента ресурса *Airplane_2* по результатам матчинга с *Sender_4*

Agent Airplane resource_3 structure					
Matchers					
Partner	Status	Name	Value	State	Type
Sender demand_5	Passive	Airplane date of dispatch	03.04.2007 16:00:00	Assigned	Simple
		Airplane Direction	1	Assigned	Simple
		Airplane free volume	800	Assigned	Scripted
		Airplane volume	800	Assigned	Simple
		Airplane load	0	Assigned	Simple
		Rent fee	15.00p.	Assigned	Scripted
		Sender volume	700	Assigned	Partner
		Course_ \$	25.00p.	Assigned	Simple
		Airplane Company	UT	Assigned	Simple

Рис. 41. Структура агента ресурса *Airplane_3* по результатам матчинга с *Sender_5*

Окончательные результаты матчинга можно видеть на рисунке 42. В результате матчинга были выполнены следующие операции резервирования:

- *Sender_demand_1* – *Aiplane_resource_1*;
- *Sender_demand_2* – *Aiplane_resource_1*;
- *Sender_demand_3* – *Aiplane_resource_1*;
- *Sender_demand_4* – *Aiplane_resource_2*;
- *Sender_demand_5* – *Aiplane_resource_3*.

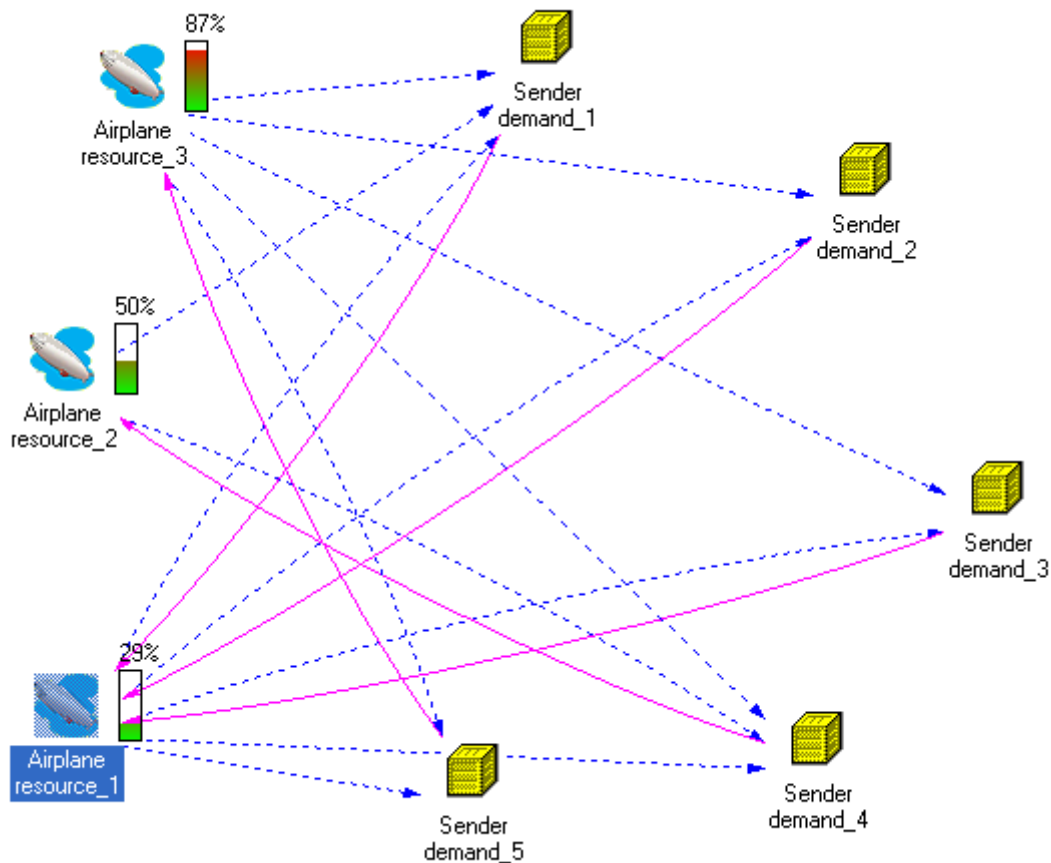


Рис. 42. Результаты матчинга

2.5.2 Отчет по результатам моделирования сцены

По результатам моделирования сцены может быть сформирован отчет с использованием команд $\langle Alt \rangle + \langle M \rangle$ или *Tools -> Make report* (рис. 43). Для генерации отчета можно выбрать наиболее информативные атрибуты с помощью фильтра атрибутов (рис. 44). Маска фильтра сохраняется и может быть загружена при повторной генерации отчета.

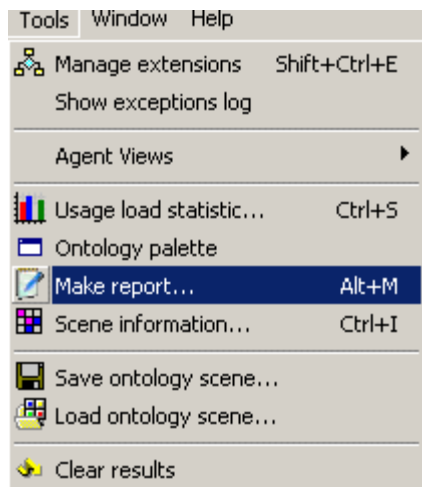


Рис. 43. Пункт меню генерации отчета

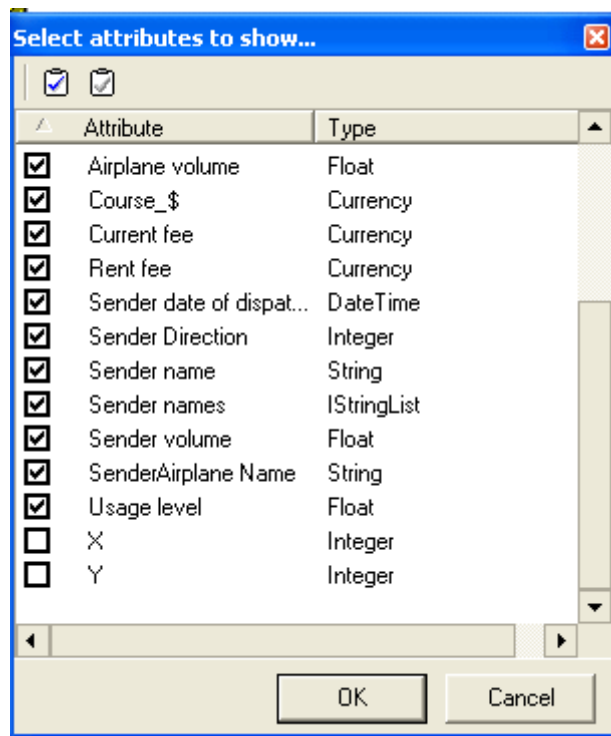


Рис. 44. Фильтр атрибутов в отчете

Отчет по сцене представлен на рисунке 45. Поля агентов имеют следующее назначение:

- *Airplane name* (идентификатор самолета);
 - *Airplane volume* (предельная масса перевозимого груза на борту, т.е. максимальная загрузка);
 - *Airplane load* (текущая загрузка самолета, кг);
 - *Airplane free volume* (свободная грузоподъемность самолета, кг);
 - *Usage level* (загрузка самолета в процентах от максимальной загрузки);
 - *Rent fee* (тариф за перевозку 1 кг груза);
 - *Sender names* (список идентификаторов отправителей, грузы которых размещены на самолете);
 - *Airplane account* (суммарная плата, полученная самолетом от всех отправителей, чьи грузы он перевозит);
 - *Airplane direction* (направление);
 - *Airplane company* (авиакомпания);
 - *Airplane date of dispatch* (время вылета).
-
- *Sender name* (идентификатор отправителя);
 - *Sender volume* (потребная масса груза, который необходимо перевезти);
 - *Current fee* (плата отправителя зарезервированному самолету);
 - *SenderAirplane Name* (идентификатор зарезервированного самолета, на котором будет перевезен груз отправителя);
 - *Sender direction* (направление перевозки);

- *Sender date of dispatch* (время и дата отправки груза).


C:\OntCons_UniIntf\Reports\SceneReport_229.htm Переход

Airplane												
Name	Airplane volume	Airplane load	Airplane free volume	Usage level	Rent fee	Airplane name	Sender names	Airplane Account	Airplane Direction	Airplane Company	Course_ \$	Airplane date of dispatch
B737	5500.00	1600.00	3900.00	29.09	14.00р.	B737	Sender_1, Sender_2, Sender_3	22 400.00р.	1	E5	25.00р.	03.04.2007 20:00:00
TY-134_1	1000.00	500.00	500.00	50.00	14.25р.	TY-134_1	Sender_4	7 125.00р.	1	SU	25.00р.	03.04.2007 8:00:00
TY-134_2	800.00	700.00	100.00	87.50	15.00р.	TY-134_2	Sender_5	10 500.00р.	1	UT	25.00р.	03.04.2007 16:00:00

Sender						
Name	Sender volume	Current fee	Sender name	Sender Airplane Name	Sender Direction	Sender date of dispatch
Sender_1	100.00	1 400.00р.	Sender_1	B737	1	03.04.2007 6:00:00
Sender_2	800.00	11 200.00р.	Sender_2	B737	1	03.04.2007 10:00:00
Sender_3	700.00	9 800.00р.	Sender_3	B737	1	03.04.2007 16:00:00
Sender_4	500.00	7 125.00р.	Sender_4	TY-134_1	1	03.04.2007 6:45:00
Sender_5	700.00	10 500.00р.	Sender_5	TY-134_2	1	03.04.2007 6:30:00

Рис. 45. Отчет по сцене

2.5.3 Сохранение сцены виртуального мира

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



- ◆ Сохраните сцену под именем *Scene_Shared Airplanes*. 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. Определите понятие разделяемого ресурса.
2. Как определить параметр «уровень использования агента ресурса» в сцене?
3. Какой флаг необходимо установить у концепта ресурс, чтобы он рассматривался при моделировании сцены как разделяемый?
4. Что такое матчер? Какие функции он выполняет?
5. Как определить задание матчеру на расчет атрибутов?
6. Для чего матчер использует обработчики событий? Какие типы событий определены для отношения матчинга?
7. Как определить скрипт, который будет выполняться при наступлении

- событий? Какие параметры у этого скрипта необходимо задать?
8. В чем особенности алгоритмов принятия решений, реализуемых машиной принятия решений с использованием метода главного условия и метода средневзвешенной оптимизации? Как установить необходимый метод?
 9. Как задать весовые коэффициенты для условий принятия решений, используемых в процессе матчнга?
 10. Как задать в сцене атрибут типа “Date Time”?
 11. Как сгенерировать отчет по результатам моделирования сцены?
 12. Какие атрибуты агентов представлены в отчете по результатам матчнга «Заказ – Разделяемый ресурс»?

4 ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Авиа-компания	Самара – Москва	Самара – Санкт-Петербург	Самара – Новосибирск
SU	Min платеж - 6.32\$ 0.53\$/kg до 45 кг 0.47\$/kg до 100 кг 0.42\$/kg до 300 кг 0.37\$/kg свыше 300 кг + 0.15\$/kg топливный сбор + 0.05\$/kg сбор за безопасность	-	-
UT	Min оплачиваемый вес 5 кг 15 руб/кг	-	-
E5	Min оплачиваемый вес 25 кг 17 руб/кг до 50 кг 14 руб/кг свыше 50 кг	Min оплачиваемый вес 25 кг 21 руб/кг до 50 кг 17 руб/кг свыше 50 кг	Min оплачиваемый вес 25 кг 25 руб/кг до 50 кг 22 руб/кг свыше 50 кг
PL	-	Min платеж – 396 руб 16.1 руб/kg до 45 кг 14.5 руб/kg до 100 кг 13.2 руб/kg до 250 кг 12.1 руб/kg до 500 кг 11.0 руб/kg до 1000 кг 9.7 руб/kg свыше 1000 кг	-
U6	-	-	Min платеж –

			880 руб 33 руб/kg до 45 кг 25.3 руб/kg до 100 кг 24.2 руб/kg СВЫШЕ 1000 кг
X8	-	-	Min оплачиваемый вес 10 кг 26 руб/кг до 45 кг 25 руб/кг до 300 кг 23 руб/кг СВЫШЕ 300 кг

```

begin
  if "_Airplane Direction" = 1 then
    begin

      if "_Airplane Company" = 'SU' then
        begin
          if ("_Sender volume"<=11.93) then
            Result:= "_Course_$" * 6.32 / "_Sender volume"
          else if ("_Sender volume">11.93) and
            ("_Sender volume"<=45)
            then Result:=(0.53+0.15+0.05)* "_Course_$"
          else if ("_Sender volume">45) and
            ("_Sender volume"<=100)
            then Result:=(0.47+0.15+0.05) * "_Course_$"
          else if ("_Sender volume">100) and
            ("_Sender volume"<=300)
            then Result:=(0.42+0.15+0.05) * "_Course_$"
          else if ("_Sender volume">300)
            then Result:=(0.37+0.15+0.05) * "_Course_$"
          end
        end

      else if "_Airplane Company" = 'UT' then
        begin
    
```

```

        if ("_Sender volume"<=5) then Result:=5*15 / "_Sender
volume"
        else Result:=15
        end

else if "_Airplane Company" = 'E5' then
begin
    if ("_Sender volume"<=25) then Result:=25*17 /
"_Sender volume"
    else if ("_Sender volume">25) and
        ("_Sender volume"<=50) then Result:=17
    else Result:=14
    end
end

else if "_Airplane Direction" = 2 then
begin

    if "_Airplane Company" = 'E5' then
        begin
            if ("_Sender volume"<=25) then Result:=25*21 /
"_Sender volume"
            else if ("_Sender volume">25) and
                ("_Sender volume"<=50) then Result:=21
            else Result:=17
            end

        else if "_Airplane Company" = 'PL' then
            begin
                if ("_Sender volume"<=8.2) then Result:=396 /
"_Sender volume"
                else if ("_Sender volume">8.2) and
                    ("_Sender volume"<=45) then Result:=16.1
                else if ("_Sender volume">45) and
                    ("_Sender volume"<=100) then Result:=13.2
            end
        end
    end
end

```

```

else if ("_Sender volume">100) and
  ("_Sender volume"<=250) then Result:=12.1
else if ("_Sender volume">250) and
  ("_Sender volume"<=500) then Result:=11
else if ("_Sender volume">500) then Result:=9.7
end
end

else if "_Airplane Direction" = 3 then
begin

  if "_Airplane Company" = 'E5' then
  begin
    if ("_Sender volume"<=25) then Result:=25*25 /
    "_Sender volume"
    else if ("_Sender volume">25) and
      ("_Sender volume"<=50) then Result:=25
    else Result:=22
  end

  else if "_Airplane Company" = 'X8' then
  begin
    if ("_Sender volume"<=10) then Result:=10*20 /
    "_Sender volume"
    else if ("_Sender volume">10) and
      ("_Sender volume"<=45) then Result:=20
    else if ("_Sender volume">45) and
      ("_Sender volume"<=300) then Result:=19
    else if ("_Sender volume">300) then Result:=17
  end

  else if "_Airplane Company" = 'U6' then
  begin
    if ("_Sender volume"<=26.7) then Result:=880 /
    "_Sender volume"

```

```
else if ("_Sender volume">26.7) and
  ("_Sender volume"<=45) then Result:=33
else if ("_Sender volume">45) and
  ("_Sender volume"<=100)then Result:=25.3
else if ("_Sender volume">100) then Result:=24.2
end

end

end;
```