

3541-10-РПР-1

СИТИС: РесТек 1.00

Руководство программиста

Редакция 1

05.12.2016

Аннотация

Данный документ является руководством программиста программы «СИТИС: РесТек». В документе приведены термины и определения, описаны шаблоны и скрипты агентов.

Авторское право

© ООО «СИТИС», 2016 г.

ООО «СИТИС» предоставляет право бесплатных печати, копирования, тиражирования и распространения этого документа в сети Интернет и локальных и корпоративных сетях обмена электронной информацией. Не допускается взимание платы за предоставление доступа к этому документу, за его копирование и распечатывание. Не разрешается публикация этого документа любым другим способом без письменного согласия ООО «СИТИС».

Оглавление

Аннотация.....	2
Авторское право	2
1. Термины и определения	4
1.1. Общие	4
1.2. Время	4
1.3. Объекты моделирования	4
1.4. Свойства агента	5
1.5. Координаты агента.....	5
1.6. Визуализация агента	6
1.7. Растровая решетка	6
1.8. Путь агента	6
1.9. Обход	6
1.10. Сервис обмена сообщениями.....	7
1.11. Отладка моделирования	7
2. Запуск программы РесТек с ключами.....	7
3. Ресурсы	8
4. Шаблон агента	8
5. Скрипт агента.....	8
6. Расшифровка номера подвижного агента.....	10
7. Визуализация агента.....	10
8. Создание отчетов	11
9. Сервис обмена сообщениями	11
9.1. Запрос	11
9.2. Послание.....	11
10. Параметры объектов моделирования	11

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1.1. Общие

- 1.1.1. **Моделирование** – (визуальное) представление запрограммированного процесса, исследование которого позволяет получить о нем некую информацию.
- 1.1.2. **Проект** – список структур данных объектов, участвующих в моделировании. Хранится в файле формата JSON, с расширением rsc.
- 1.1.3. **Интерпретатор** - покомандный исполнитель кода.
- 1.1.4. Среда интерпретатора – уникальная для каждого агента (или обладателя скрипта) область памяти, в которой обеспечены независимость и целостность данных.
- 1.1.5. **JS API** – библиотека функций JavaScript, интерпретируемых в программе РесТек.
- 1.1.6. **Скрипт** - последовательность функций JS API, определяющая действия объектов моделирования.
- 1.1.7. **Простая функция** - функция, которая выполняет одно действие.
- 1.1.8. **Специализированная функция** – функция, которая выполняет несколько последовательных типовых операций.

1.2. Время

- 1.2.1. **Глобальное время** – момент реального календарного времени по Гринвичу.
- 1.2.2. **Системное время** – момент времени, соответствующий дате и времени дня, предоставленный операционной системой. При правильной настройке операционной системы должен соответствовать глобальному времени с поправкой на часовой пояс.
- 1.2.3. **Модельное время** – дискретная величина, представляющая время моделируемого процесса активности реальных объектов, моделируемых агентами.
- 1.2.4. **Шаг моделирования** – временной интервал между отдельными дискретными моментами времени моделирования.
- 1.2.5. **Начало моделирования** – начальный момент времени моделирования, в котором все агенты инициализированы и находятся в начальном состоянии. Этот момент времени моделирования принимается равным нулю.
- 1.2.6. **Момент модельного времени** – одно из дискретных значений времени моделирования в диапазоне от начального значения до текущего.
- 1.2.7. **Текущее модельное время** – момент времени, для которого осуществляется расчет текущих параметров среды и агентов и определяются действия агентов на текущем шаге цикла моделирования.
- 1.2.8. **Продолжительность моделирования** – продолжительность времени от начала моделирования до окончания моделирования, когда все агенты достигли своих целей.
- 1.2.9. **Расчетное время активности агента** – время от начального состояния агента до конечного состояния (момента достижения цели).
- 1.2.10. **Продолжительность расчета** – продолжительность времени для выполнения расчета на компьютере. Определяется как разница между системным временем начала и окончания расчета. Зависит от загрузки компьютера другими задачами.
- 1.2.11. **Продолжительность расчетного процесса** (процессорное время расчета) - время от начала расчета до конца расчета, в течение которого процессор обрабатывал расчетную задачу. Не зависит от загрузки процессора другими задачами.

1.3. Объекты моделирования

- 1.3.1. **Сцена** – совокупность всех объектов, участвующих в расчете.
- 1.3.2. **Супервайзер** – надсмотрщик, контролер процесса моделирования. В его задачи входит запуск моделирования с установленными параметрами необходимое количество раз с помощью скрипта.
- 1.3.3. **Решатель** – исполнитель, реализатор процесса моделирования. Содержит такие параметры, как шаг моделирования, максимальную продолжительность моделирования и др.
- 1.3.4. **Трассировщик** - пошаговый исполнитель процесса моделирования. Выполняет заданный скрипт в каждый модельный момент времени, кратный шагу трассировки.

- 1.3.5. **Уровень** - плоская поверхность, ограниченная многоугольным (прямоугольным) контуром, по которому могут двигаться агенты.
- 1.3.6. **Агент** – объект, взаимодействующий с другими агентами и окружающей средой при моделировании. Взаимосвязь выполняется через функции JS API, описанные в скрипте агента. Проигрывание скрипта или его функций происходит в среде интерпретатора JavaScript – уникальной для каждого агента, что обеспечивает независимость и целостность данных.
- 1.3.7. **Тип агента** – скаляр, относящий агента к соответствующей группе, которая определяет его поведение при моделировании.

В проекте реализованы следующие типы:

- 1.) **Препятствие** – неподвижный агент, непроходимый для других агентов на все время моделирования, тип 100-199
- 2.) **Участок пути** - неподвижный агент, в скрипте которого запрограммирован расчет плотности в регионе, представленном границей агента с помощью алгоритма растровой заливки, тип 200-210
- 3.) **Локация** – неподвижный агент, который может являться целью для подвижного агента, тип 300-399
- 4.) **Подвижный** – агент, меняющий свое местоположение на сцене, тип 500-599

1.4. Свойства агента

- 1.4.1. **Представление агента** - контейнер данных, в котором определены параметры агента, логика действий агента (скрипт), и визуальное представление агента.
- 1.4.2. **Параметры агента** – данные, которые однозначно задают агента при его создании.
- 1.4.3. **Скрипт агента** – интерпретируемая программа на JS, в которой реализована логика поведения агента (функция инициализации, рисования, шаговая и т.д.)
- 1.4.4. **Идентификатор агента** – целочисленный параметр, задающий однозначное соответствие между числом и агентом, присваивается агенту при создании.
- 1.4.5. **Шаблон агента** – набор данных, которые используются для присвоения начальных параметров при создании агента в интерфейсе программы, путем копирования из реализации шаблона.
- 1.4.6. **Реализация шаблона агента** – JSON-файл, в нем перечислены свойства объекта, не относящиеся к местоположению на уровне - информация о форме и логике. Форма – полигон, его вершины заданы в декартовой системе координат с началом в точке (0,0). Логика – адрес файла со скриптом.
- 1.4.7. **Форма агента** – 2D полигон в декартовой системе координат агента – проекция объема агента на горизонтальную плоскость пола.
- 1.4.8. **Граница агента** – 2D полигон, полученный в результате преобразования координат (параллельный перенос и поворот системы) формы агента в аналитические координаты уровня.
- 1.4.9. **Центр агента** – точка на плоскости, является началом локальной системы координат агента, в которой задается его геометрия.
- 1.4.10. **Угол поворота агента** – угол по часовой стрелке, на который повернута локальная система координат агента на сцене, определяет направление движения агента.
- 1.4.11. **Верх агента** – задает максимальное положение агента по оси Oz.
- 1.4.12. **Низ агента** – задает минимальное положение агента по оси Oz.
- 1.4.13. **Состояние агента** – числовой код, меняющийся во времени, характеризующий набор параметров агента, соответствующих различным существенно отличным состояниям моделируемого реального объекта. Как правило выделяется три основных состояния – начальное состояние (0), нормальное движение (спокойная ходьба -1), достижение цели (2).
- 1.4.14. **Личное пространство** – параметр, задающий минимальное расстояние, на которое агент может подойти к преграде.

1.5. Координаты агента

- 1.5.1. **Аналитические координаты** – пара действительных чисел, задающих положение точки в декартовой системе координат относительно левого нижнего угла уровня (начала с.к.) в метрах, первое число в паре – координата по горизонтали (по оси Oх), второе - по вертикали (по оси Oy).
- 1.5.2. **Дискретные координаты** – пара целых чисел, задающая номера ряда и колонки ячейки растровой решетки сцены. Крайняя левая нижняя ячейка имеет дискретные координаты (0,0).

1.6. Визуализация агента

- 1.6.1. **3D модель** – описание изображения агента в пространстве. Реализация - файл формата 3DS, OBJ или STL.
- 1.6.2. **Идентификатор 3D модели** – целочисленный параметр, задающий однозначное соответствие параметра и 3D модели, к нему нельзя применять арифметический операции, нужно только передавать в качестве параметра в функции рисования.
- 1.6.3. **Визуализация агента** – набор параметров, данных, упорядоченный набор 3D моделей агента, представляющих визуальное представление различных кадров движения агента в разных состояниях с различной детализацией. Последовательное изображение фаз агента которых анимирует плавное шаговое движение агента в текущем состоянии.
- 1.6.4. **Клип** – визуальное представление движение агента за некоторый интервал времени, последовательное проигрывание которого представляет анимированное движение агента в заданном состоянии.
- 1.6.5. **Номинальное время клипа** – усредненный интервал времени, соответствующий движению агента в клипе (при усредненной скорости движения для рассматриваемого состояния).
- 1.6.6. **Шаг клипа** – длина пути агента во время клипа (для неподвижных состояний равен 0).
- 1.6.7. **Фаза клипа** – последовательное отображениями кадров в клипе через равные промежутки времени
- 1.6.8. **Кадр клипа** – отображение 3D объекта, соответствующего данной фазе клипа. Разным фазам может соответствовать один и тот же 3D объект.
- 1.6.9. **Представление клипа** – набор фаз клипа.
- 1.6.10. **Уровень детализации** – условная характеристика качества визуального представления клипа (1-минимальный уровень).

1.7. Растровая решетка

- 1.7.1. **Растровая решетка** - прямоугольная двумерная матрица заданного размера, состоящая из ячеек, для которой задана прямоугольная система координат. Начало координат соответствует нижнему левому углу решетки.
- 1.7.2. **Растр** - элемент растровой решетки, для простоты квадратный, определенного размера. Каждая ячейка может принимать состояние от 0 до 1000.
- 1.7.3. **Состояние ячейки** – параметр, который задает значение в ячейке (пустая, граница региона, препятствие, агент).
- 1.7.4. **Карта путей** – растровая решетка, в ячейках которых хранится время пути от нее до локации и предыдущая ячейка кратчайшего пути.
- 1.7.5. **Радиус огибания препятствий для расчета карты путей** – величина, на которую расширяются границы статических препятствий, чтобы подвижные агенты не пересекали препятствия.

1.8. Путь агента

- 1.8.1. **Текущий путь агента** – траектория перемещения агента от начально до текущего положения.
- 1.8.2. **Расчетный путь агента** - путь агента от начального положения до конечного положения – достигнутой цели или конечного состояния.
- 1.8.3. **Целевой путь агента** – кратчайшая траектория движения агента до цели.
- 1.8.4. **Карта путей локации** – карта с длиной целевых путей до локации, рассчитанная для каждой точки сцены, не заданных препятствиями.
- 1.8.5. **Вес целевого пути** – скалярная величина, характеризующая целевой путь агента, вычисляемая для сравнения близости к достижению цели разными агентами. По умолчанию равна длине текущего целевого пути агента.

1.9. Обход

- 1.9.1. **Динамическое препятствие** – подвижный агент, который находится на пути.
- 1.9.2. **Обходной путь** – дискретный путь, рассчитанный при обнаружении на пути агента динамического препятствия, предотвращающий их столкновение/пересечение.

1.10. Сервис обмена сообщениями

- 1.10.1. **Сервис обмена сообщениями** (почтовый сервис) - специализированная служба обмена данными между агентами. Хранит послания в базе данных.
- 1.10.2. **Сообщение** – данные, которые агент может передавать другим агентам через сервис обмена сообщениями движка.
- 1.10.3. **Запрос** (request) – сообщение, требующее возвращения ответа. Действия агента, направившего сообщение, приостанавливается до получения ответа. (Функция не возвращает результат до формирования ответа запрошенным агентом).
- 1.10.4. **Послание** (message) – сообщение, не требующее получения ответа. Действия агента, направившего сообщение, продолжаются. (Функция возвращает управление сразу, не дожидаясь реакции адресата)
- 1.10.5. **Адресат** – идентификатор агента, являющегося адресатом сообщения.
- 1.10.6. **Текст сообщения** – суть сообщения, данные (строка) для передачи. В зависимости от подтипа сообщения преобразуется (конвертируется) к необходимому формату.
- 1.10.7. **Тип сообщения** – код, классифицирующий сообщение для обработки почтовым сервисом и адресатом. Зарезервировано на будущее. По умолчанию 0.
- 1.10.8. **Подтип сообщения** – код, классифицирующий сообщение для обработки адресатом. Зарезервировано на будущее. По умолчанию 0. Может использоваться для простой передачи численного параметра вместо обработки строки текста сообщения.

1.11. Отладка моделирования

- 1.11.1. **Точки останова** – объекты, реализующие преднамеренное прерывание выполнения моделирования в режиме отладки в заданное время или промежуток времени, и совершение определенных действий.
- 1.11.2. **Ролик (видеоролик)** – слайд шоу кадров состояния сцены с визуализированными агентами, с заданными параметрами камеры для заданного промежутка времени.
- 1.11.3. **Карта** – изображение скалярных свойств ячеек растровой решетки с заданным масштабом в цветовом эквиваленте.
- 1.11.4. **Масштаб карты** – задает соответствие одной растровой ячейки количеству пикселей.
- 1.11.5. **Шкала карты или колорбар** – цветной столбец с определенными минимальным и максимальным значениями, который задает соответствие скалярной величины и цвета.
- 1.11.6. **Карта плотностей** – карта, в которой скалярным свойством ячеек является плотность.
- 1.11.7. **Диаграмма двойная** – изображение значений ячеек (определенных) растровой решетки в виде кругов, размер и цвет которых показывают две скалярные величины.
- 1.11.8. **Шкала диаграммы** - сайзбар – столбец цветных кругов, который задает соответствие величин цвету и радиуса круга.
- 1.11.9. **Окно отладчика** – окно, в которое выводятся результаты выполнения скриптов.
- 1.11.10. **Номер окна отладчика** – порядковое число окна отладчика при создании. Если номер отрицательный, то окно - буфер рисования, и является невидимым.

2. ЗАПУСК ПРОГРАММЫ РЕСТЕК С КЛЮЧАМИ

- 2.1. Запуск программы РесТек выполняется со следующими ключами:

- -s=имя_файла_скрипта – присваивает супервайзеру скрипт
- -r – открывает последний проект
- -m – запускает расчет
- -t – после расчета программа завершает работу

- 2.2. Примеры:

```
"C:\Program Files\Sitis\restech\ResTech.exe" "C:\Program Files\Sitis\restech\BottleTest.rsc"  
-s="C:\Scripts\SuperVisorScript.js" -m -t  
"C:\Program Files\Sitis\restech\ResTech.exe" -r -s="C:\Scripts\SuperVisorScript.js" -m -t
```

3. РЕСУРСЫ

- 3.1. Настройка путей до используемых данных в JSON-файле tune.txt (расположен с файлом запуска программы РесТек):
- "SCRIPTS" – путь до папки, в которой находятся все скрипты агентов, по умолчанию ...\\restech\\resource\\Scripts
 - "TEMPLATE" – путь до папки, в которой находятся все шаблоны агентов, по умолчанию ...\\restech\\resource\\AgentTemplate
 - "IMAGE" – путь до папки, в которую производится автоматическое сохранение карт/диаграмм из окна отладчика, по умолчанию ...\\restech\\resource\\Maps
 - "DOCLAB" – путь до установленной программы ДокЛаб, по умолчанию C:\\Program Files\\Sitis\\ДокЛаб
 - "VISUAL" – путь до папок визуализации агентов, по умолчанию ...\\restech\\resource\\Visual

4. ШАБЛОН АГЕНТА

- 4.1. Шаблон агента должен иметь имя *.atpl.txt и следующий формат JSON:

```
{
  "OBJ": "AgentTemplate",
  "NAME": "011М-Здоровая женщина", //имя, отображаемое в списке шаблонов
  "DESCRIPTION": "Здоровая женщина в летней одежде", //пояснения
  "FORM": //форма агента
  {
    "SIZE": 10, //количество вершин
    "POINTS": [[0.05,0.15],[0.15,0.13],[0.23,0.05],[0.23,-0.05],[0.15,-0.13],
               [-0.15,-0.13],[-0.23,-0.05],[-0.23,0.05],[-0.15,0.13],[-0.05,0.15]]
    //вершины, координаты в метрах в системе агента
  },
  "FORMBORDER": [[0,0,0],0], //перо границы
  "FORMFILL": [[0,50,205],1,7], //кисть
  "TOP": 1.7, //верх агента
  "BOT": 0, //низ агента
  "PEN1": [[0,0,0],0],
  "PEN2": [[0,0,0],0],
  "FILL1": [[0,50,205],1,6],
  "FILL2": [[0,50,205],1,6],
  "PARAM": [1.0, 0], //доп.параметры
  "LP": 0, //личное пространство
  "POLYGON1": {"POLSIZE": 10,
               "POLPOINTS": [[0.05,0.15],[0.15,0.13],[0.23,0.05],[0.23,-0.05],[0.15,-0.13],
                             [-0.15,-0.13],[-0.23,-0.05],[-0.23,0.05],[-0.15,0.13],[-0.05,0.15]]},
  "POLYGON2": {"POLSIZE": 10,
               "POLPOINTS": [[0.05,0.15],[0.15,0.13],[0.23,0.05],[0.23,-0.05],[0.15,-0.13],
                             [-0.15,-0.13],[-0.23,-0.05],[-0.23,0.05],[-0.15,0.13],[-0.05,0.15]]},
  "SCRIPT": "011М_Здоровая_женщина.txt", //скрипт агента
  "TA": 500, //тип агента
  "VISUAL": "011М-Здоровая_женщина_V3D" //визуализация агента
}
```

5. СКРИПТ АГЕНТА

- 5.1. Требования – размер не должен превышать 12 МБ, и 200 символов в каждой строке.
- 5.2. Обязательно отсутствие синтаксических ошибок.

- 5.3. Возможно использование многострочного `/**/` и однострочного `//` комментариев, операций и операторов `+`, `-`, `*`, `/`, `=`, `!=`, `==`, `<`, `>`, `>=`, `<=`, `||`, `&&`, `++`, `--`. Комментарии только с новой строки.
- 5.4. Определены следующие математические функции:
- `Math.abs(a)` – возвращает абсолютное значение числа (модуль)
 - `Math.round(a)` – возвращает ближайшее целое число
 - `Math.min(a, b)` – возвращает минимальное из чисел
 - `Math.max(a, b)` – возвращает максимальное из чисел
 - `Math.range(x, a, b)` – возвращает число `a`, если `x < a`, `b`, если `x > b` и `x`, если `a < x < b`
 - `Math.sign(a)` – возвращает знак числа (`-1` – отрицательный, `0` – `0`, `1` – положительный)
 - `Math.PI()` – возвращает число `PI`
 - `Math.toDegrees(a)` – возвращает значение числа, переведенного из радиан в градусы
 - `Math.toRadians(a)` – возвращает значение числа, переведенного из градусов в радианы
 - `Math.sin(a)` – возвращает синус угла в радианах
 - `Math.asin(a)` – возвращает арксинус угла в радианах
 - `Math.cos(a)` – возвращает косинус угла в радианах
 - `Math.acos(a)` – возвращает арккосинус угла в радианах
 - `Math.tan(a)` – возвращает тангенс угла в радианах
 - `Math.atan(a)` – возвращает арктангенс угла в радианах
 - `Math.sinh(a)` – возвращает гиперболический синус угла в радианах
 - `Math.cosh(a)` – возвращает гиперболический косинус угла в радианах
 - `Math.tanh(a)` – возвращает гиперболический тангенс угла в радианах
 - `Math.atanh(a)` – возвращает гиперболический арктангенс угла в радианах
 - `Math.E()` – возвращает число Непера (основание натурального логарифма)
 - `Math.log(a)` – возвращает логарифм по основанию `a`
 - `Math.log10(a)` – возвращает десятичный логарифм
 - `Math.exp(a)` – возвращает экспоненту в степени `a`
 - `Math.pow(a,b)` – возвращает `ab`
 - `Math.sqr(a)` – возвращает `a2`
 - `Math.sqrt(a)` – возвращает квадратный корень из `a`
- 5.5. Определены функции работы со строками, целыми числами и массивами:
- `Math.rand()` – возвращает произвольное число
 - `Math.randint(min, max)` – возвращает целое число из диапазона
 - `charToInt(ch)` – конвертирует строку в целое число
 - `String.indexOf(search)` – поиск подстроки в строке, если не найдено – вернет `-1` (доступ `->`)
 - `String.substring(lo,hi)` – возвращает символы строки, начиная с `lo` по `hi` (доступ `->`)
 - `String.charAt(i)` – возвращает `i`-тый символ строки (доступ `->`)
 - `String.charCodeAt(i)` – возвращает код `i`-того символа (доступ `->`)
 - `String.fromCharCode(char)` – возвращает символ по коду (доступ `->`)
 - `String.split(separator)` – возвращает массив строк, полученных в результате разбиения строки по сепаратору (доступ `->`)
 - `Integer.parseInt(str)` – преобразует строку в целое число
 - `Integer.valueOf(str)` – преобразует символ в целое число
 - `Array.contains(obj)` – `1`, если объект содержится в массиве, `0` – если нет. (доступ `->`)
 - `Array.remove(obj)` – удаляет объект из массива (доступ `->`)
 - `Array.join(separator)` – соединяет в строку все элементы массива, разделяя сепаратором. Если аргумент не задан, элементы будут соединены запятыми. (доступ `->`)
- 5.6. В среде интерпретатора объявлены следующие объекты:
- `NVW` – идентификатор последнего созданного невидимого окна отладчика
 - `C2D(nx, ny)`, `P2D(x,y)`, `P3D(x,y,z)` (см. документацию JS API)
- 5.7. В среде интерпретатора агента объявлены и инициализированы следующие глобальные переменные:
- `ID` – идентификатор агента
 - `agentPLGN` – идентификатор полигона, задающий границу агента

- 5.8. В скрипте агента в обязательном порядке должны быть описаны следующие функции:
- Init() – функция инициализации агента
 - Step() – шаговая функция
 - Draw2D() – функция рисования агента на плоскости
 - Draw3D() – функция рисования агента в пространстве
 - reply(id_from, value, type, subtype) – функция обработки запроса
 - Trace2D(DWIN, K) – функция, которую вызывает трассировщик в своем скрипте, выполняет отрисовку агента в отладочном окне с идентификатором DWIN по ключу K.
- 5.9. В скрипте трассировщика в обязательном порядке должны быть описаны следующие функции:
- TraceStart() – функция, выполняемая до начала моделирования
 - Trace() – функция, выполняемая в каждый модельный момент времени, кратный шагу трассировки
 - TraceEnd() – функция, выполняемая после завершения моделирования
- 5.10. Особое внимание:
- В реализации интерпретатора языка JavaScript не предусмотрено возвращение результата в виде объекта C2D, P2D и P3D, только в виде массива размером 1, в котором нулевым элементом является объект.
 - Для изучения сохраненной истории данных об агентах, необходимо на каждом шаге расчета устанавливать с свойства агента такие вычисляемые параметры, как плотность и скорость.
 - Моделирование не закончится, пока у всех агентов не будет состояние достижения цели (2).

6. РАСШИФРОВКА НОМЕРА ПОДВИЖНОГО АГЕНТА

- 6.1. Номер подвижного агента имеет следующий вид – XYZM, где:
- 6.2. X – пол агента
- 0 – женский
 - 1 – мужской
- 6.3. Y – возраст агента
- 0 – пожилой
 - 1 – взрослый
 - 2 – подросток
 - 3 – ребенок
- 6.4. Z – порядковый (отличительный) номер по группе мобильности
- 0 – пожилой
 - 1 – здоровый
 - 2 – слепой
 - 3 – с двумя опорами
 - 4 – с одной опорой
 - 5 – в кресле
 - 6 – пожарный
 - 7 – пожарный с ДГЗ
- 6.5. M – визуализация 3д-моделями
- 6.6. Специализированные агенты, например, собака, не имеют номеров.

7. ВИЗУАЛИЗАЦИЯ АГЕНТА

- 7.1. Для трехмерного изображения агента можно использовать функции рисования примитивов (см. JS API), либо клип.

- 7.2. Пусть $S1$ – шаг клипа в текущем состоянии, $P1$ – длина пути агента в текущем состоянии, $P2$ – текущий путь в клипе, $F1$ – номер текущей фазы, $N1$ – количество фаз в клипе.
- 7.3. Тогда, чтобы отобразить 3д-модель с нужным номером текущей фазы в функции `Draw3D()` вычислим:
- 7.4. $P2 = P1 - \lfloor \text{целая часть числа } (P1/S1) * S1 \rfloor$;
- 7.5. $F1 = N1 * P2 / S1$;

8. СОЗДАНИЕ ОТЧЕТОВ

- 8.1. Для создания автоматических отчетов программа «РесТек» использует программу «СИТИС: ДокЛаб».
- 8.2. Чтобы использовать функции записи инструкций и генерации отчета необходимо в настроечном файле `tune.txt`, расположенном рядом с exe-файлом РесТека, проверить путь до установленной программы ДокЛаб в строке "DOCLAB". Если программа ДокЛаб не установлена или путь не совпадает, пользоваться указанными функциями запрещено.

9. СЕРВИС ОБМЕНА СООБЩЕНИЯМИ

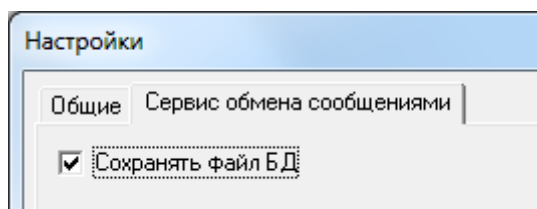
9.1. Запрос

- 9.1.1. Чтобы агент смог обработать направленный ему запрос, необходимо реализовать в его скрипте функцию `reply`.
- 9.1.2. В зависимости от текста запроса функция возвращает необходимые данные.
- 9.1.3. Пример функции `reply` агента – участок пути:

```
function reply(id_from, value, type, subtype)
{
    if (value == "D")
    {
        var retD = getD(ID);
        return retD;
    }
}
```

9.2. Послание

- 9.2.1. Т.к. послания не требуют получения ответа, их хранение реализовано в SQLite БД. Если в настройках стоит флажок «Сохранять файл БД», для каждого запуска расчета будет создан файл с именем `имя_проекта_год_месяц_дата_часы_минуты_сек_мс.dblite`, в котором сохранены все послания агентов за время моделирования.



- 9.2.2.
- 9.2.3. Если флажок не установлен, послания хранятся в файле `MessagingService.dblite`, который создается при установке РесТек и перезаписывается на каждом расчете.

10. ПАРАМЕТРЫ ОБЪЕКТОВ МОДЕЛИРОВАНИЯ

- 10.1. Супервайзер, Решатель и Агенты имеют свойство параметры объекта. Это строка, содержащая инициализированные переменные, разделенные точкой с запятой. Супервайзер переопределяет параметры Решателя, Решатель переопределяет параметры всех Агентов.