

РАЗРАБОТКА ПОДСИСТЕМЫ РАСЧЕТА ВЫБРОСА ЗАГРЯЗНЯЮЩИХ ВЕЩЕСТВ В АТМОСФЕРУ АВТОТРАНСПОРТНЫМИ СРЕДСТВАМИ

А. Н. Маренков

Воронежский государственный университет

Поступила в редакцию 01.03.2010 г.

Аннотация. В статье описывается разработка подсистемы расчета объема выброса загрязняющих веществ (ЗВ) в атмосферу автотранспортными средствами (АС). Приведены основные структуры и классы, используемые подсистемой. Описана общая схема процедуры расчета. В принципе построения подсистемы закладывается возможность разделения основных функций между параллельными процессами на одной или разных вычислительных машинах.

Ключевые слова: загрязняющие вещества, автотранспортные средства, разработка подсистемы

Abstract. The article discusses a vehicle atmospheric emission pollutant calculation subsystem design. General structures and classes used by subsystem are given. General arrangement of emission pollutant calculation is discussed. In principle of subsystem building are laid possibility of partitioning main functions between parallel processes, that executed by single or different computers.

Key words: atmospheric pollutant, vehicle, subsystem development

ВВЕДЕНИЕ

Для минимизации ущерба, причиняемого человечеством окружающей среде, необходимо иметь возможность максимально точно рассчитать объем загрязняющих веществ (ЗВ), попадающих в окружающую среду в процессе той или иной деятельности человека.

Также необходимо знать, каким образом этот объем распределяется в пространстве, чтобы минимизировать воздействие ЗВ на жилые зоны, сельскохозяйственные участки и прочее.

Частным случаем загрязнения окружающей среды является выброс ЗВ в атмосферу. Для этого типа загрязнения существуют определенные нормативные рекомендации по расчету, и выработаны формы для сбора информации об источниках загрязнения атмосферы (ИЗА). Поэтому для этой области экологических расчетов существует класс унифицированных программ расчета загрязнения атмосферы (УПРЗА).

Многие ИЗА работают не постоянно, а в определенные временные промежутки. Кроме того, в эти промежутки ИЗА может функционировать с определенной вероятностью. Такие источники получили название «нестационарные ИЗА».

Автотранспортные средства (АС) также выбрасывают ЗВ в атмосферу. При этом АС не только функционируют в определенные временные промежутки, но и меняют свое местоположение с течением времени. Таким образом, АС можно назвать «передвижными нестационарными ИЗА».

Однако существующие УПРЗА не рассчитаны на работу с передвижными нестационарными ИЗА. Следовательно, они не могут обеспечить максимальную достоверность расчетов. В связи с этим актуальна задача разработки подсистемы расчета выброса ЗВ в атмосферу автотранспортными средствами.

В статье [1] был приведен язык описания временных данных АС, и было дано математическое обоснование метода дискретизации временных интервалов функционирования АС. В статьях [3, 4] было определено, что наиболее эффективным методом представления АС является моделирование с использованием многоагентной системы [5].

Таким образом, необходимо определить, как эти подходы будут воплощены при разработке непосредственно подсистемы расчета выброса ЗВ в атмосферу автотранспортными средствами. Термин «подсистема» здесь используется потому, что расчет выброса включается в более

крупную систему, предоставляющую возможности ввода данных, отображения результатов и разделения расчетных операций между отдельными процессами и отдельными вычислительными машинами.

1. ОСНОВНЫЕ СТРУКТУРЫ.

Как было показано в [4], основными элементами модели, с определенными упрощениями, являются:

- агенты (автотранспортные средства);
- область действия агентов;
 - отрезки дорог;
 - площадки;
 - светофоры;
- ключевые точки.

Агенты (CAgent) в данной модели представляют собой автотранспортные средства. Согласно [4] агенты обладают следующими ключевыми характеристиками:

- тип (легковые, грузовые, автобусы) – представляется в виде перечислимого типа – Type: EAgentType {Car, Cargo, Bus};
- вид топлива (бензин, ДТ, сжиженный газ, сжатый газ) – представляется в виде перечислимого типа – FuelType: EFuelType {Petrol, DieselFuel, CondensedGas, CompressedGas};
- экологический класс (Евро 0 – 3) – представляется в виде перечислимого типа – Class: EEcoClass {Euro0, Euro1, Euro2, Euro3}.
- группа (смысловое объединение агентов) – Group: ubyte (> 0);

Отрезки дорог (CSegment) представляют собой некий прямой участок дороги. Для упрощения мы будем оперировать только прямыми отрезками. Таким образом, все не прямые участки дорог будут аппроксимированы набором прямых отрезков. Согласно [4] отрезки дорог обладают следующими ключевыми характеристиками:

- идентификатор участка – ID: ulong (первый бит – «0»);
- состояние (работает, не работает) – Work: bool;
- координаты концов отрезков – X1, Y1, X2, Y2: double[4];
- количество полос движения от первой точки отрезка (X1, Y1) ко второй (X2, Y2) – ForwardLaneCount: ubyte;
- количество полос движения от второй точки отрезка (X2, Y2) к первой (X1, Y1) – BackwardLaneCount: ubyte;

- длина отрезка в автотранспортных средствах (принимается усредненная длина в 2 м.) – Length: ushort;
- массив связей первого конца отрезка – Links1: ulong[];
- массив связей второго конца отрезка – Links2: ulong[];
- вероятность функционирования (может быть как статической вероятностью, так и функцией от параметров окружения).

Изначально отрезок задается через координаты концов. Поля Length, Links1, Links2 иницируются пустыми значениями. Однако перед началом моделирования и проведения расчета координаты концов пересчитываются в длину отрезка и заполняются массивы связей.

Тип длины отрезка выбран ushort, так как средняя протяженность прямых участков дорог в городе составляет менее 10 км. Этот тип данных дает запас значений в 12 раз.

Массивы связей определяют, с чем соединены соответствующие концы отрезка. Массивы связей заполняются таким образом, что первые два бита используются для определения, к какому участку соответствующей сущности подсоединен данный конец. Остальные биты задают идентификатор сущности (отрезка или площадки).

Площадки (CArea) представляют собой некую прямоугольную область, внутри которой агенты могут перемещаться в любых направлениях. Обладают следующими основными параметрами:

- идентификатор площадки – ID: ulong (первый бит – «1»);
- координаты серединной линии – X1, Y1, X2, Y2: double[4];
- ширина площадки в автотранспортных средствах – Width: ushort;
- длина площадки в автотранспортных средствах – Length: ushort;
- входные точки площадки – EntryPoint: Struct[4];
 - индекс по длине – LNum: ulong;
 - индекс по ширине – WNum: ulong;
 - связи этой входной точки – Links: ulong[].

Изначально площадка задается через координаты серединной линии и ширины. Поля Length и EntryPoint иницируются пустыми значениями. Однако перед началом моделирования и проведения расчета координаты сре-

динной линии пересчитываются в длину площадки и заполняются массивы входных точек.

Для упрощения мы принимаем, что входных точек площадки не может быть больше 4 – поэтому в связи для указания участка сущности используется только два бита.

Массив входных точек определяет, на какую ячейку площадки автотранспортное средство попадает при въезде, или с какой ячейки оно может покинуть площадку.

Светофоры (CTrafficLights) – регулирующие элементы. Могут располагаться только в точках соединения отрезков. Один светофор может регулировать движение только для двух отрезков. Основные параметры:

- идентификатор первого отрезка – SegmentID: ulong;
- идентификатор второго отрезка – SegmentID: ulong;
- временные интервалы работы;
- вероятности нахождения в каждом из трех состояний (зеленый свет, красный свет, не работает);

Ключевая точка (CKeypoint) – точка, несущая смысловую информацию для агентов. Обладает следующими существенными параметрами:

- координаты – X, Y: double[2];
- массив оценок;
 - группа – Group: ubyte;
 - оценка – Mark: float;
- идентификатор сущности привязки – LinkID: ulong;
- индекс по длине – LNum: ulong;
- индекс по ширине – WNum: ulong;

Изначально точка задается через координаты. Поля LinkID, LNum и WNum инициализируются пустыми значениями. Однако перед началом моделирования и проведения расчета через координаты разрешается структурный элемент, содержащий эту точку. И определяется положение точки на конкретном элементе.

Точка может нести как положительную (в этом случае агенты будут стремиться к ней), так и отрицательную (агенты будут стремиться от нее) оценку. Точки могут иметь одинаковый смысл для всех агентов (Group = 0), или варьировать свою оценку для разных подмножеств агентов.

2. УПРАВЛЯЮЩИЕ ОБЪЕКТЫ

Для организации моделирования и расчета необходимо ввести дополнительные управляю-

щие объекты. Эти объекты будут совершать определенные операции над основными структурами.

Исходные данные задаются с привязкой к картографической информации. Все объекты области действия агентов обладают набором координат (географических или метрических, с привязкой к определенной локальной системе координат). Однако для моделирования эта информация не имеет значение. При моделировании используются связи между объектами. Для перевода картографических координат в связи используется класс CLinker.

Для осуществления моделирования необходимо вводить агентов в область действия агентов. Для этого используется класс CTrafficGenerator. Данный класс обладает большим количеством функций и настроек, чтобы обладать возможностью ввода агентов в область согласно гибким и сложным правилам. Для этого класса можно задать правила для определения участка области, на котором будет осуществляться ввод агентов, временные схемы ввода агентов, процентные соотношения для ввода агентов разных групп и т.д.

В идеале, каждый агент должен самостоятельно вычислять путь своего движения, опираясь на данные, которые он может определить самостоятельно (окружение в потоке, средняя скорость движения потока и т.п.). Однако на практике это сопряжено с высокой сложностью и избыточностью расчета. Поэтому мы будем использовать отдельный класс, который будет перемещать агентов по области действия. Тем не менее, этот класс – CTrafficController – при принятии «решения» будет опираться исключительно на данные, доступные конкретному агенту. Объектов данного класса может быть несколько, и они могут контролировать смежные непересекающиеся участки области действия. Неконтролируемых участков быть не может.

В процессе движения агенты выделяют загрязняющие вещества. Объем и состав этих выделений зависит от параметров агента. Расчетом этих параметров занимается класс CContaminantCalculator. Он производит расчет в конкретной ячейке области действия агентов в момент, когда агент, занимающий эту ячейку, меняет режим своего функционирования или перемещается.

По результатам, рассчитанным CContaminantCalculator, класс CIsolinesBuilder произво-

дит построение изолиний и изоконтуров в конкретном участке области действия в конкретный момент времени.

3. ПРОЦЕДУРА РАСЧЕТА ВЫБРОСА ЗАГРЯЗНЯЮЩИХ ВЕЩЕСТВ

На входе подсистема получает заданные картографические отрезки и площадки области действия агентов. Также входные данные содержат настройки для объекта класса *CTrafficGenerator*, определяющие правила генерации агентов.

Далее *CLinker* осуществляет проход по отрезкам, площадкам и ключевым точкам, заполняя пустые параметры связей.

После этого начинается моделирование. На каждом шаге *CTrafficGenerator* осуществляет выброс заданного количества агентов в область действия. Для вновь добавленных агентов *CTrafficController* генерирует маршрут к целевой для них точке. С определенной долей вероятности маршрут может быть проложен некорректно или неоптимально.

Для существующих агентов *CTrafficController* определяет возможность дальнейшего движения, если агент находится на прямом участке дороги. Если агент находится перед светофором, то определяется возможность его движения и направление движения. При достижении ключевой точки агентом, *CTrafficController* либо выводит его из зоны действия, либо генерирует новый маршрут согласно параметрам агента.

В случае если объектов класса *CTrafficController* несколько, то при достижении агентом границы своего действия, один объект *CTrafficController* передает информацию об агенте другому объекту, на чью территорию перемещается агент. Границы объектов класса *CTrafficController* могут проходить исключительно по прямым участкам дорог (не по площадкам и не по перекресткам). Это связано с тем, что разные объекты класса *CTrafficController* могут обрабатываться разными процессорами, или находиться на разных машинах в одной сети, и поэтому они должны быть максимально независимы и не иметь дублирующих данных.

В процессе изменения местоположения агента или режима его функционирования, вызываются функции объекта *CContaminantCalculator*. Объект класса *CContaminantCalculator* производит расчет выброса загрязняющего

вещества и заносит значение в хранилище данных с меткой местоположения выброса и временного интервала, на котором происходил этот выброс. Согласно общей концепции подсистемы данные о выбросе могут передаваться на отдельную машину, и там же храниться и обрабатываться.

Построение результирующих изолиний и изоконтуров [2] может осуществляться объектом класса *CIsoLinesBuilder* либо параллельно моделированию, либо после него по расчетным данным, находящимся в хранилище данных. Также это может производиться параллельно, но на другой машине, которая объединена с расчетной машиной в сеть.

ЗАКЛЮЧЕНИЕ

В статье были показаны основные структуры и объекты, используемые при проведении расчета выброса ЗВ. Также была дана общая схема расчета и построения результирующих изолиний и изоконтуров.

Согласно принципам, заложенным в [2, 3, 4] данная подсистема должна в полной мере использовать преимущества распределенных вычислений. В данной статье мы не касались объектов, ответственных за передачу и синхронизацию данных между разными процессами или компьютерами, но ясно видно, что принцип построения подсистемы позволяет осуществлять ключевые операции независимо друг от друга – параллельно.

СПИСОК ЛИТЕРАТУРЫ

1. Маренков А. Н. Учет нестационарности функционирования источников загрязнения атмосферы при проведении расчета рассеивания // Вестник Воронежского государственного университета. Серия «Системный анализ информационных технологий». – Воронеж, 2009. – № 1. – С. 95–99.
2. Маренков А. Н. Построение изолиний и изоконтуров на поле результатов расчета выброса загрязняющих веществ // Вестник Воронежского государственного университета. Серия «Системный анализ информационных технологий». – Воронеж, 2008. – № 2. – С. 127–130.
3. Маренков А. Н. Учет автотранспортных средств при проведении расчета рассеивания выбросов загрязняющих веществ в атмосферу // Вестник Воронежского государственного технического университета. Том 5. – Воронеж, 2009. – № 9. – С. 82–84.
4. Маренков А. Н. Моделирование и расчет выброса загрязняющих веществ в атмосферу автотранспортными средствами с использованием многоагентной системы // Вестник Воронежского государс-

твенного университета. Серия «Системный анализ информационных технологий». – Воронеж, 2009. – № 2. – С. 127–130.

Маренков Алексей Николаевич – аспирант кафедры программирования и информационных технологий факультета компьютерных наук Воронежского Государственного Университета. Тел. 8 (960) 120-08-47. E-mail: amarenkov@inbox.ru.

5. *Kirn S. Multiagent Engineering / S. Kirn, O. Herzog, P. Lockemann, O. Spaniol.* – M.: Heidelberg, 2006. – 617 с.

Marenkov Alexey Nikolaevich – Post-Graduate student The dept. of the Programming and Information Technologies, Computer Science Faculty, Voronezh State University Tel. 8 (960) 120-08-47. E-mail: amarenkov@inbox.ru.