

УДК 519.688

Э.С. Соколова, И.Е. Волкова, О.П. Тимофеева, Е.С. Кадиленко

**ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ СИСТЕМЫ УПРАВЛЕНИЯ  
БЕСПИЛОТНЫМИ ТРАНСПОРТНЫМИ СРЕДСТВАМИ**

Нижегородский государственный технический университет им. Р.Е. Алексеева

Рассматриваются основные особенности программной реализации системы управления беспилотными транспортными средствами в пределах ограниченной сети автодороги. Представлены результаты работы системы на небольшом наборе данных и выполнен анализ полученных результатов.

*Ключевые слова:* автопилот, интеллектуальная система управления, программная реализация, транспортные средства.

В последнее время появляется всё больше интеллектуальных систем помощи водителю, а также ведутся активные разработки полностью автопилотируемых транспортных средств. Наиболее ярким представителем автопилотируемых транспортных средств в настоящее время является беспилотный автомобиль Google. Есть основания полагать, что в будущем подобные автопилотируемые транспортные средства практически полностью заменят привычные всем автомобили, и появится возможность эффективного управления всем транспортным потоком. Подобная система, позволяющая оптимизировать перемещения всех автопилотируемых транспортных средств, рассматривается в [1]. Для проверки эффективности данной системы было разработано клиент-серверное приложение.

Клиент-серверное приложение состоит из двух частей: клиент, представляющий собой отдельное приложение с графическим интерфейсом для конфигурирования графа, и сервер, который является консольным приложением, содержащим непосредственно реализацию алгоритма. Взаимодействие клиента с сервером осуществляется через сеть посредством сокетов. Для этого создана отдельная библиотека, реализующая проприетарный протокол обмена данными между клиентом и сервером.

Сервер разработан как многопоточное консольное приложение с четырьмя потоками: поток соединения с клиентом, поток вычисления карты оптимальных путей, поток вычисления оптимальных маршрутов и буферный поток.

Поток соединения с клиентом обрабатывает входящие запросы установления связи с клиентом, принимает конфигурацию графа и перенаправляет её в остальные потоки, а также отправляет клиенту результаты каждой промежуточной итерации нахождения оптимальных путей. Поток соединения с клиентом реализован с помощью двух классов (рис. 1).

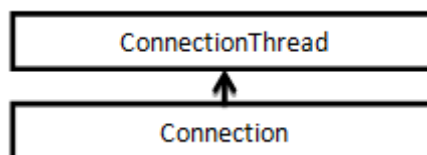
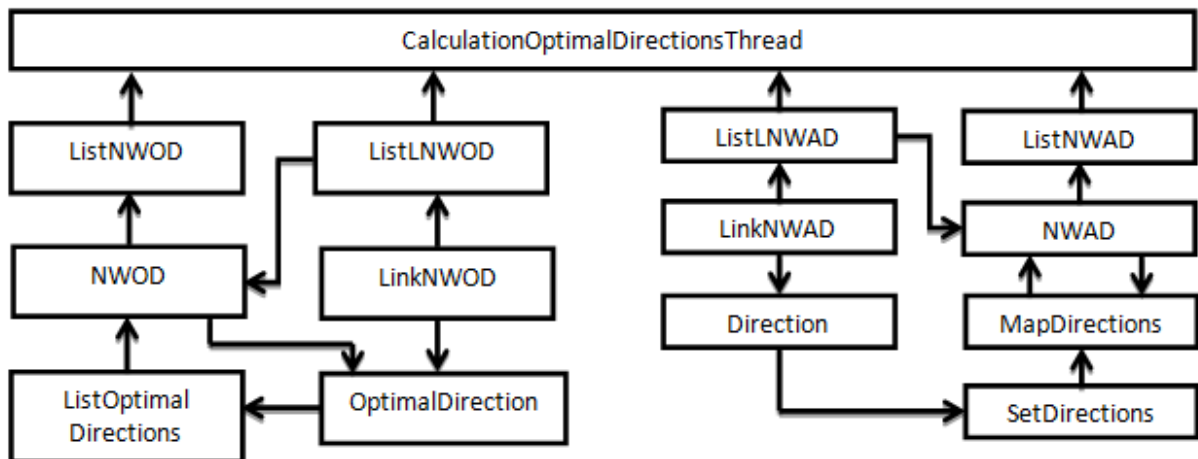


Рис. 1. Структура классов для потока соединения

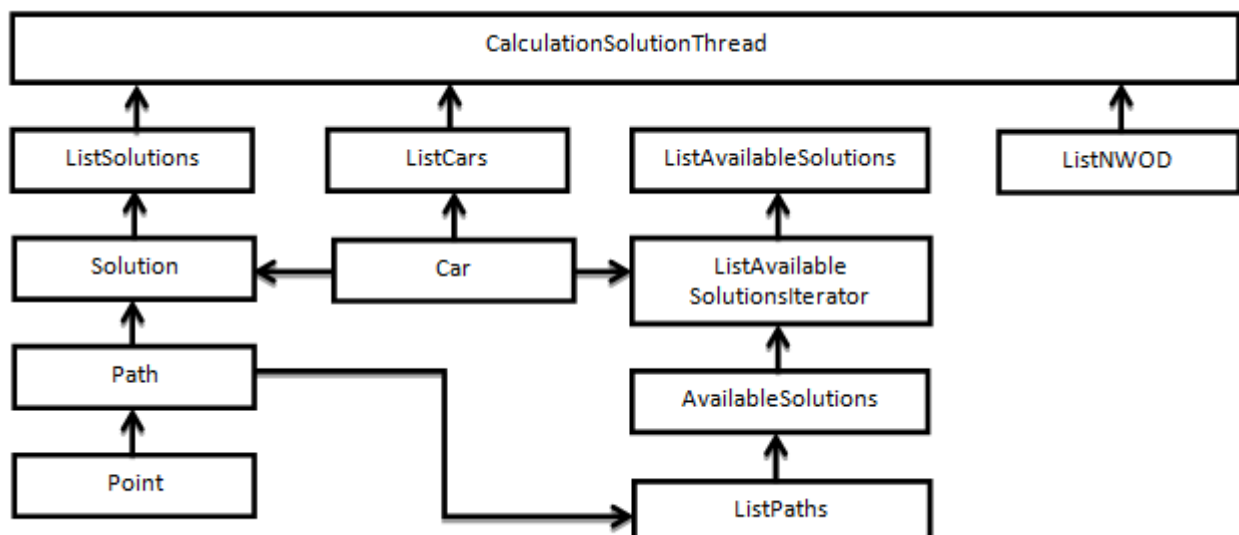
Поток вычисления карты оптимальных путей содержит конфигурацию графа, представляемую в виде двух списков: списка узлов и списка связей. Реализовано два типа узлов и связей. Первый тип узлов и связей используется для хранения начальной конфигурации графа, а также используется в процессе создания карты оптимальных путей.

Списки, составленные по этим узлам и связям, создаются один раз и хранятся до завершения работы приложения. Второй тип узлов и связей используется для хранения карты оптимальных путей. Списки второго типа создаются как результат вычисления оптимальных путей и посылаются на хранение в буферный поток. Реализация потока вычисления карты оптимальных путей состоит из 14 классов (рис. 2).



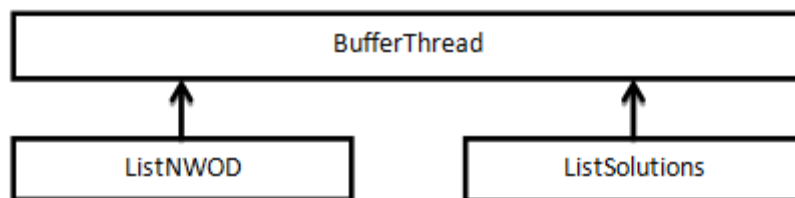
**Рис. 2. Структура классов для потока вычисления карты оптимальных путей:**  
*NWAD – Node With All Directions; NWOD – Node With Optimal Directions; LNWAD – Link Node With All Directions; LNWOD – Link Node With Optimal Directions*

Поток вычисления оптимальных маршрутов содержит список транспортных средств и указатель на список узлов второго типа, представляющий карту оптимальных путей. В этом потоке работает таймер, при срабатывании которого запрашивается указатель на текущую карту оптимальных маршрутов. Период срабатывания таймера равен промежутку времени, за которое вычисляется промежуточное решение. После получения указателя начинается вычисление промежуточного решения для транспортных средств. Результат вычисления представляется четырьмя типами объектов: точкой, путем, решением и списком решений. Точка является минимальной структурой решения и содержит три параметра: идентификатор узла, скорость и время. Путь является списком точек. Решение объединяет транспортное средство и путь для него. Результатом работы алгоритма является список решений, содержащий промежуточные пути для всех транспортных средств, которые ещё не достигли конечной точки (рис. 3).



**Рис. 3. Структура классов для потока вычисления оптимальных маршрутов**

Буферный поток используется для хранения текущей карты оптимальных путей и хранения необработанных решений. Этот поток необходим для синхронизации работы потока вычисления оптимальных маршрутов и потока вычисления карты оптимальных путей (рис. 4).



**Рис. 4. Структура классов для буферного потока:**  
*NWOD – Node With Optimal Directions*

Тестирование системы проводилось на двух компьютерах находящихся в одной сети. Сервер работал на стационарном компьютере с 4-ядерным процессором IntelCorei5 с частотой 2.80 Ghz, оперативной памятью общим объёмом 4 Гб, под управлением операционной системы Ubuntu 14. Клиент работал на ноутбуке с 4-ядерным процессором IntelCorei5 с частотой 2.50 Ghz, оперативной памятью общим объёмом 6 Гб, под управлением операционной системы Windows 7.

Для тестирования использовался граф, содержащий 246 вершин и 256 связей. Суммарная длина дороги, смоделированная данным графом, составляет примерно 500 м и имеет две полосы движения, по одной полосе в каждом направлении. В процессе тестирования перед каждым запуском добавлялось по одному транспортному средству. Максимальное количество транспортных средств, одновременно участвующих в движении, составило 38.

Результаты представлены на графике, отображающем зависимость между количеством машин в конфигурации и наибольшим временем нахождения промежуточного решения (рис. 5).



**Рис. 5. Диаграмма зависимости максимального времени вычисления промежуточного решения от количества машин на графе**

Как видно из диаграммы, система успешно работает (время просчета решения меньше максимально допустимого), если число транспортных средств, одновременно участвующих в движении, не превышает 37.

Следует отметить, что тестирование велось на относительно слабом оборудовании для данного типа задач. Ожидается, что повышение вычислительной мощности процессора даст значительно лучшие результаты работы системы управления движением. Также отметим,

что программно реализован упрощённый метод разрешения конфликтов, оптимизация которого позволила бы значительно ускорить нахождение промежуточных решений. Помимо этого дальнейшее распараллеливание вычислений с целью оптимизации используемых ресурсов вычислительного оборудования поможет сократить время нахождения промежуточных решений.

В целом достигнутый результат можно считать положительным. Рассмотренная система является хорошей базой для дальнейших исследований возможности оптимизации движения множества транспортных средств в заданном координатном пространстве путей.

- 
1. **Соколова, Э.С.** Интеллектуальная система управления беспилотными транспортными средствами в пределах ограниченной сети автодороги / Э.С. Соколова, О.П. Тимофеева, Е.С. Кадиленко // Научно-технический вестник Поволжья. 2014. №4. С. 191–193.

*Дата поступления  
в редакцию 11.12.2014*

**E.S. Sokolova, I.E. Volkova, O.P. Timofeeva, E.S. Kadilenko**

## **PECULIARITIES OF CONTROL SYSTEM SOFTWARE FOR UNMANNED VEHICLES**

Nizhny Novgorod state technical university n.a. R.E. Alexeev

The main peculiarities of control system software for unmanned vehicles within a limited road network is described in this paper. The results of modeling with the small amount of data are presented. Analyses of the results are described in this paper.

*Key words:* autopilot, intelligent control system, software implementation, vehicles.