

Материал опубликован в сборнике докладов третьей Всероссийской научно-технической конференции по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД-2007). СПб.: 2007. Т. 1, с. 296–301.

Применение автоматного программирования для имитационного моделирования разезда машин на нерегулируемом перекрестке равнозначных дорог

А. С. Никитин, М. Ю. Чураков, А. А. Шалыто

В настоящее время большинство автошкол России не оборудованы компьютерами. Поэтому все объяснения инструктора выполняются у доски. В результате на занятиях рассматриваются лишь несколько типичных ситуаций, возникающих в процессе дорожного движения, и объясняется алгоритм поведения водителя в этих ситуациях. Однако ситуации, которые возникают на практике, зачастую оказываются значительно сложнее и требуют от водителя быстрого анализа дорожной обстановки и принятия решения. В таких случаях у водителя нет времени для того, чтобы перебрать в памяти все рассмотренные на уроках примеры и вспомнить, как необходимо действовать. Задача обучения состоит как раз в том, чтобы сформировать в сознании учащегося ясный алгоритм принятия решения в любой дорожной обстановке и довести навыки управления транспортным средством до автоматизма. Для этого авторы предлагают внедрять в автошколы моделирующие системы, с помощью которых учащиеся смогут самостоятельно имитировать интересующие их дорожные ситуации.

В данной работе рассматривается один из наиболее сложных аспектов правил дорожного движения (ПДД): правила проезда перекрестков, в частности, нерегулируемых перекрестков равнозначных дорог. В ПДД имеется два пункта, относящиеся к этому вопросу:

- 13.11 На перекрестке равнозначных дорог водитель безрельсового транспортного средства обязан уступить дорогу транспортным средствам, приближающимся справа (Это правило в просторечии называется правилом «помехи справа»).
- 13.12 При повороте налево или развороте водитель безрельсового транспортного средства обязан уступить дорогу транспортным средствам, движущимся по равнозначной дороге со встречного направления прямо или направо.

Кроме этого, известно, что если же ни один их участников не может начать движение из-за того, что кто-то другой препятствует этому, имея больший приоритет, то водителям необходимо договориться (например, подав звуковой сигнал) между собой о порядке разезда на перекрестке.

Изучив учебные пособия по этому вопросу, авторы пришли к выводу, что в некоторых ситуациях задача определения очередности проезда является нетривиальной. Рассмотрим, например, ситуацию, изображенную на рис. 1. Здесь задача состоит в том, чтобы машина, находящаяся справа выполнила поворот налево, а остальные машины проехали прямо. Несмотря на то, что для каждой машины существует помеха справа, порядок проезда однозначно определен: сначала машина справа выезжает на середину и останавливается, затем проезжает машина снизу, потом машина слева, и, наконец, машина, остановившаяся на середине, завершает поворот налево.

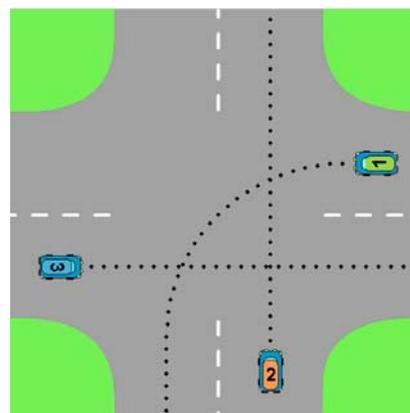


Рис.1. Машина справа поворачивает налево, машины снизу и слева едут прямо.

Таким образом, правило проезда перекрестка, казалось бы, можно сформулировать следующим образом: каждая машина должна двигаться до пересечения с траекторией движения машины, приближающейся справа, а затем, пропустив её, продолжить движение. Однако такая формулировка может привести к ситуации взаимной блокировки машин. Например, если с каждой стороны перекрестка движется по одной машине прямо, то они все выедут на середину перекрестка и не смогут продолжить движение.

Кроме того, в ПДД отсутствуют указания о выборе конкретной траектории движения машины, хотя, например, при повороте налево или развороте траектория будет зависеть от направлений движения других машин. Поэтому водитель должен ясно понимать, как ему необходимо двигаться по перекрестку, и в каком месте он должен остановиться, пропуская другие машины.

Постановка задачи

Создать модель разъезда машин на нерегулируемом перекрестке равнозначных дорог. При этом предполагается, что перекресток имеет квадратную форму и с каждой его стороны может двигаться не более одной машины в любом из четырех направлений: направо, прямо, налево, на разворот.

Модель должна удовлетворять следующим требованиям:

1. Не должно быть взаимных блокировок.
2. Не должно быть аварий.
3. Очередность разъезда и траектории движения машин в типовых случаях должны соответствовать примерам из учебных пособий.
4. Число случаев, когда водители должны договариваться между собой о порядке разъезда, должно быть минимальным. Это означает, что во всех ситуациях, когда существует способ разъезда, который не нарушает правило «помехи справа», машины должны выполнить это правило.
5. Модель должна быть имитационной и представлять собой мультиагентную систему – **каждая машина должна принимать решения самостоятельно, исходя из визуальной и звуковой информации на перекрестке, доступной водителю.**

Реализация

Авторы систематизировали возможные случаи разъезда машин на рассматриваемом перекрестке и разработали универсальное правило проезда перекрестка для водителя автомобиля. Таким правилом стал конечный автомат, однозначно определяющий правильное поведение водителя в любой дорожной обстановке.

Для реализации поставленной задачи использовалось инструментальное средство проектирования автоматных программ *UniMod* [1] и язык *Java*. Это средство позволяет создавать объектно-ориентированные программы, поведение которых описывается автоматами, взаимодействующими с объектами управления с помощью событий и входных/выходных переменных. В качестве объектов управления используются любые объекты, которые в основном выполняют не логические, а вычислительные действия. При этом каждый автомат связывается с поставщиками событий и объектами управления. Переход автомата из одного состояния в другое происходит при наступлении определенного события и выполнении логического условия, составленного из входных

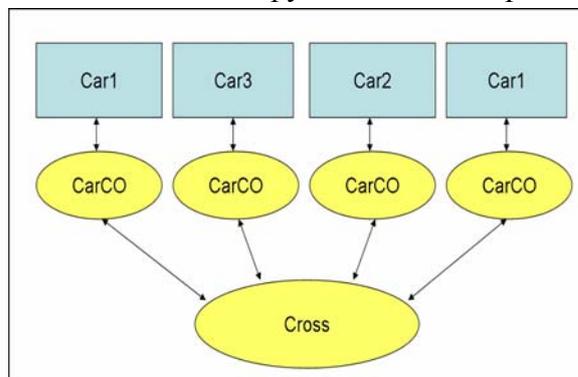


Рис. 2. Схема взаимодействия автоматов и объектов. Автоматы обозначены прямоугольниками, объекты – овалами.

переменных объектов управления. При выполнении перехода автомат может выполнять выходные воздействия – вызывать определенные методы объекта управления. Для облегчения визуального восприятия диаграмм связей и переходов автоматов, в пакете *UniMod* используются краткие обозначения событий, объектов, входных и выходных переменных, состоящие из буквы и числа (*e4*, *z10*, *x6* и т.д.).

Общая схема взаимодействия автоматов во время моделирования изображена на рис. 2. Каждой машиной управляет автомат, взаимодействующий со своим экземпляром класса *CarCO*, инкапсулирующего машину. Объект класса *CarCO* является для него одновременно и поставщиком событий, и объектом управления. Каждый объект *CarCO*, в свою очередь, взаимодействует с объектом *Cross*, отвечающим за передачу сообщений между машинами.

Таким образом, технология автоматного программирования позволяет реализовать мультиагентную систему, состоящую из набора взаимодействующих объектов, поведением которых управляют конечные автоматы.

Модель перекрестка

Территория перекрестка разбивается на 13 зон (рис. 3). Их нумерация выполняется относительно каждой конкретной машины – для каждой машины встречная машина будет появляться в зоне 1, левая – в зоне 7, правая – в зоне 6 и т.д.

После анализа возможных вариантов разъезда авторы пришли к выводу, что можно выделить три различные траектории поворота налево или разворота: при встречном разъезде правыми бортами (малая траектория), при обычном повороте или развороте (средняя траектория) и при объезде машины, которая уже выехала на середину перекрестка (большая траектория). Таким образом, каждая машина может двигаться по одной из восьми возможных траекторий: одна траектория поворота направо, одна – прямо (рис. 4), три – налево (рис. 5) и три – на разворот (рис. 6). Траектория выбирается автоматом в начале моделирования и не может быть изменена после того, как машина начала движение.



Рис. 3. Разбиение перекрестка на зоны. Машина на середине перекрестка находится в зоне 13 и пропускает машину, движущуюся вниз

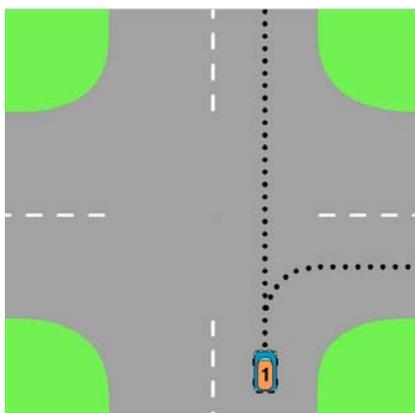


Рис. 4. Траектории движения прямо и направо

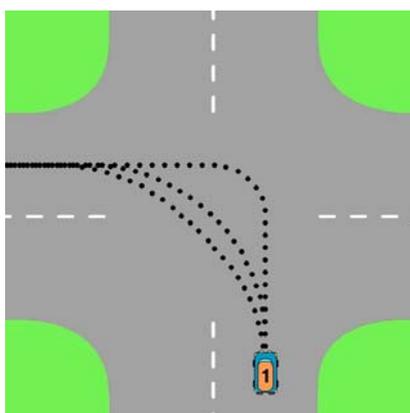


Рис. 5. Траектории поворота налево

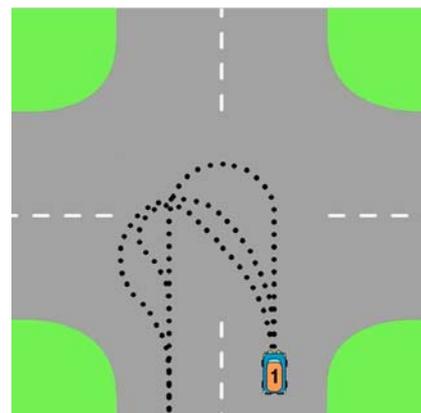


Рис. 6. Траектории разворота

Интерфейс автомата

Направление движения каждой машины задается при редактировании конфигурации перекрестка. Если выбрано направление налево или на разворот, то конкретная траектория движения (одна из трех) должна быть выбрана автоматом в начале моделирования.

Каждая траектория проходит по нескольким зонам перекрестка. Во время движения объект *CarCO* машины отслеживает номер зоны, в которой находится центр машины. При перемещении из одной зоны в другую объект *CarCO* генерирует сообщение об освобождении

зоны, которое через объект *Cross* транслируется остальным машинам. Кроме сообщений о покидании зон другими машинами, автомат получает сообщения, когда его машина пересекает одну из разделительных линий.

Моделирование начинается с того, что объект *Cross* рассылает два сообщения всем машинам:

1. «Включить указатели поворота» (сообщение *e2*). При получении этого сообщения автомат должен включить необходимые указатели поворота для того, чтобы другие машины смогли правильно оценить дорожную обстановку и выбрать тактику дальнейших действий.
2. «Начать движение» (сообщение *e4*). При получении этого сообщения автомат может выбрать траекторию движения (если машина движется налево или на разворот), и, если необходимо, машина может начать движение.

После выбора траектории автоматы взаимодействуют друг с другом с помощью передачи сообщений, которые генерируются при движении машин. Автомат может обрабатывать необходимые для принятия решения сообщения и выполнять только два выходных воздействия на своём объекте управления *CarCO*: «Ехать» и «Ждать». Для принятия решения автомат использует входные переменные объекта управления *CarCO*, которые характеризуют окружающую обстановку: расположение машин на перекрестке и состояние их указателей поворота.

В данной модели взаимная блокировка машин может возникнуть в одном из двух случаев:

- если все четыре машины двигаются прямо;
- если все четыре машины двигаются налево или на разворот.

Эта проблема решается за счёт подачи одной из машин (любой) звукового сигнала другим машинам, который означает, что эта машина намерена первой выехать на перекресток. При этом остальные машины получают сообщение *e6* «Другая машина подала сигнал» и должны её пропустить. За счёт того, что автоматы обрабатывают сообщения по очереди, не возникает конфликтной ситуации, в которой несколько машин одновременно подают звуковой сигнал.

Виды автоматов

В данном проекте были реализованы три разновидности автоматов, определяющих различное поведение машин:

1. «Правильный водитель». Этот автомат является корректной реализацией поставленной задачи. Гарантируется, что если все машины управляются автоматами этого типа, то они всегда разъедутся правильно. Диаграмма переходов данного автомата изображена на рис. 7.
2. «Сломанные указатели поворота». Этот автомат является модификацией автомата №1 «Правильный водитель», отличающейся тем, что при выполнении поворота или разворота автомат не включает указатели поворота. Машина, управляемая данным автоматом, может привести к ДТП, так как другие участники движения принимают решения, исходя из предположения, что данная машина намерена двигаться прямо.
3. «Наглый водитель». Этот автомат правильно выбирает траекторию движения и включает указатели поворота, но не пропускает машины, имеющие больший приоритет. Использование данного автомата также позволяет моделировать аварийные ситуации.

Рассмотрим первый автомат (рис. 7). При получении сообщения *e2* «Включить указатели поворота», автомат переходит в одно из трех состояний, соответствующих различным направлениям движения, и включает необходимые указатели поворота. После этого автомат ожидает получения одного из двух сообщений: *e4* «Начать движение» или *e6* «Другая машина подала сигнал». Дальнейшее поведение автомата зависит от выбранного направления движения. Наибольший интерес представляет случай поворота налево или разворота (состояние 4).

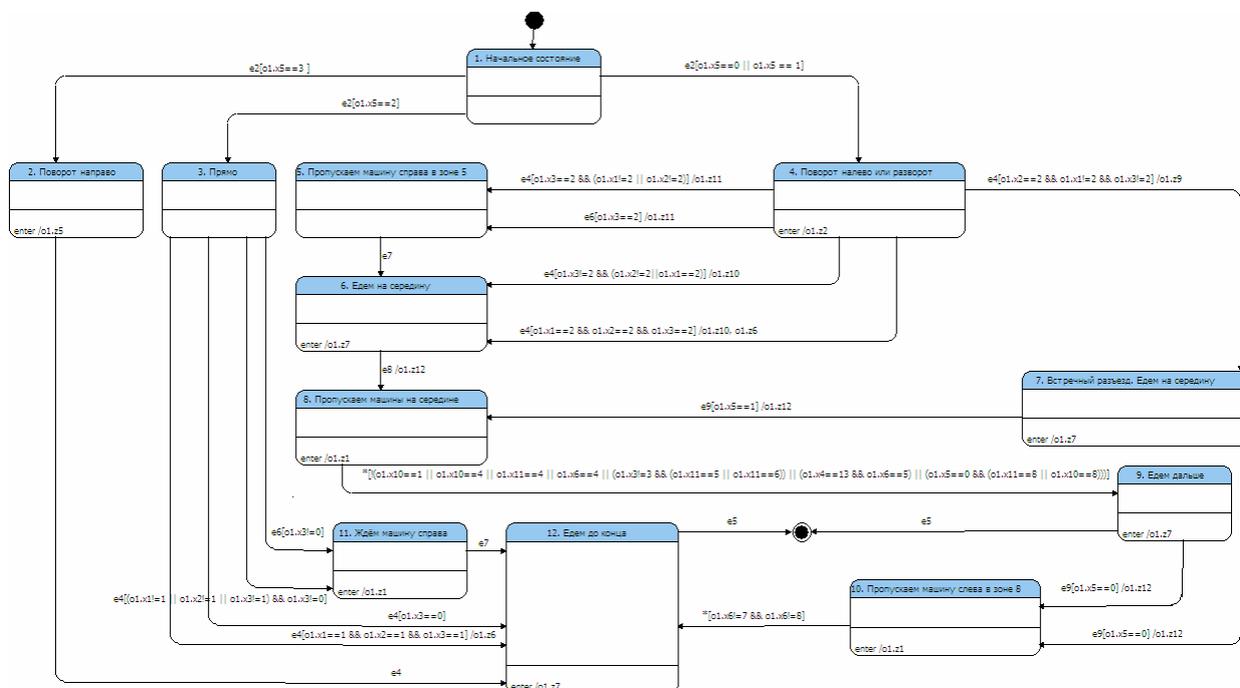


Рис. 7. Диаграмма переходов автомата №1 «Правильный водитель»

В этом случае, если встречная машина движется налево, а машина слева не движется налево, то автомат выбирает малую траекторию и разъезжается со встречной машиной правыми бортами (состояние 7). Иначе, если машина справа поворачивает налево, то необходимо её пропустить, а затем объехать её сзади по большой траектории (состояние 5). В остальных случаях автомат выбирает среднюю траекторию. При пересечении середины перекрестка автомат обрабатывает сообщение $e8$ «Машина пересекла продольную разделительную полосу» и, если необходимо, останавливается, чтобы пропустить встречную машину и машину справа (состояние 8). Если же машина разворачивается, то необходимо также пропустить машину слева, так как она становится «помехой справа» для данной машины (состояние 10). Пропустив все машины, имеющие больший приоритет, автомат переходит в состояние 12 «Едем до конца».

Моделирование завершается, когда все машины покинули перекресток, либо остановились по причине столкновения с другими машинами.

Выводы

В данной работе рассмотрена задача моделирования разъезда машин на нерегулируемом перекрестке равнозначных дорог. Эту задачу можно реализовать различными способами, но именно использование технологии автоматного программирования для построения мультиагентной системы позволяет формализовать логику принятия решения водителем.

Важным этапом проектирования автомата является описание событий, инициирующих переход автомата в новое состояние и выполнение определенных действий, входных переменных объекта управления (факторов принятия решений) и выходных воздействий (возможных действий водителя). Эта формализация позволяет сразу же обратить внимание учащегося на наиболее значимые события и факторы, определяющие действия водителя.

Модифицируя автомат, можно оценивать значения его отдельных элементов и демонстрировать возможные последствия несоблюдения или неверной трактовки правил дорожного движения. В частности, можно моделировать и анализировать ДТП.

Графическая форма представления автомата облегчает анализ алгоритма принятия решений.

Более подробная информация о проекте будет опубликована на сайте <http://is/ifmo.ru> в разделе «UniMod-проекты».

Источники

1. Гуров В.С., Мазин М.А., Нарвский А.С., Шалыто А.А. UML. Switch-технология. Eclipse //Информационно-управляющие системы. 2004. № 6. <http://is.ifmo.ru/works/uml-switch-eclipse/>