

ИМИТАЦИЯ ДВИЖЕНИЯ СУДОВ В ПРОСТРАНСТВЕННО-НЕОДНОРОДНОЙ И НЕСТАЦИОНАРНОЙ СРЕДЕ В ДИСКРЕТНО-СОБЫТИЙНОЙ ПАРАДИГМЕ МОДЕЛИРОВАНИЯ МОРСКИХ ТРАНСПОРТНЫХ СИСТЕМ

А.Г. Топаж, О.В. Таровик (Санкт-Петербург)

Введение. Наиболее распространённым подходом к созданию компьютерных имитационных моделей производственных и транспортно-логистических процессов в современных условиях является дискретно-событийное моделирование. При этом, в отличие, например, от системно-динамического подхода, в дискретно-событийной модели течение времени представляется не непрерывным, а описывается как цепочка дискретных событий, нерегулярно разбросанных по временной оси и меняющих состояние изучаемой динамической системы. Поведение системы в промежутках между событиями остаётся за рамками рассмотрения, поскольку полагается, что в этих временных интервалах всё равно «ничего не происходит». Неоспоримым преимуществом такого подхода является вычислительная эффективность, что даёт возможность имитировать динамику даже очень сложных многоагентных систем на больших временных интервалах.

Одной из базовых составляющих множества различных производственных и логистических моделей является процесс перемещения активных агентов, которые являются объектами моделирования. Примерами таких агентов являются различные транспортные средства, партии товаров и сырья, ходовые части оборудования и т.д. Перемещение, по своей сути, является непрерывным во времени процессом, т.к. скорости и координаты объекта постоянно изменяются. Для того, чтобы описать непрерывное перемещение в рамках дискретно-событийного подхода почти всегда используется следующее упрощение: движение агента из точки старта в точку назначения задаётся лишь двумя событиями – отправлением и прибытием. Логическая семантика самого движения представляет собой просто «задержку» (интервал времени заданной длительности) между этими событиями. Последнее, однако, не означает, что в пользовательском интерфейсе модели нельзя показать агент движущимся. Все современные среды имитационного моделирования (AnyLogic, ARENA, VenSim, GPSS и др.) позволяют создавать визуально непрерывные анимации для логически дискретного процесса перемещения. Однако семантически каждое такое элементарное движение (например, блок *moveTo* в процессной диаграмме среды AnyLogic) представляет собой целостный и неразделимый «чёрный ящик». Пользователь не в состоянии ни получить, ни изменить текущее состояние находящегося внутри него агента–заявки.

Такая идеология оказывается вполне пригодной для моделирования процессов, в которых агенты и ресурсы перемещаются в некой сети с предопределённой топологией (Network Based Modeling). Например, в моделях цепочки поставок автомобильным или железнодорожным транспортом. Это обусловлено тем, что в этом случае выполняются два важных условия. Во-первых, возможные траектории перемещения от начальной к конечной точке по рёбрам заданного графа (сети автомобильных или железных дорог) существуют в ограниченном количестве. Во-вторых, можно обоснованно допустить, что основные характеристики движения (скорость, расход топлива) существенно не изменяются в ходе рейса и могут быть заданы заранее и относиться ко всему маршруту в целом.

В случае морского судоходства первое условие не выполняется, т.к. суда могут двигаться по любым маршрутам внутри акватории (за исключением отдельных участков регламентированной навигации). А в случае арктического судоходства не

выполняется и второе условие, поскольку лёд на высокоширотных маршрутах отличается существенной пространственной неоднородностью (сложные ледовые участки постоянно перемежаются участками слабого льда или чистой воды), а также высокой степенью временной нестационарности (картина торосов и ледовых сжатий может кардинально изменяться всего за несколько часов). Поэтому задание в имитационной модели предопределённого маршрута и постоянных характеристик движения (скорость или время перехода) является в данном случае недопустимым упрощением, искажающим всю характерную специфику исследуемой системы. Далее в статье описан подход, который используется нами для того, чтобы преодолеть эти противоречия, то есть адекватно описать перемещение судов ледового плавания и ледоколов в имитационных моделях морских арктических транспортных систем, оставаясь при этом в рамках концепции дискретно-событийного моделирования.

Материалы и методы. Задача состоит в том, чтобы адекватно отразить в стандартной динамической имитационной модели перемещение транспортного средства от заданной точки отправления к заданной точке назначения в неоднородной и нестационарной непрерывной пространственной среде. При этом должны быть обеспечены две принципиальные возможности:

- определение наиболее рационального маршрута как перед началом движения (т.е. на весь маршрут), так и непосредственно в ходе перемещения (оперативная корректировка оставшегося участка частично пройденного маршрута);
- определение текущих параметров движения в текущей точке маршрута, исходя из складывающихся в этой точке условий окружающей среды и ходовых качеств моделируемого транспортного средства.

Поскольку для достижения поставленных целей необходимо «влезть внутрь» описания глобального процесса перемещения, то в качестве решения нами было выбрано расширение стандартного блока движения путём реализации нового универсального компонента «неравномерное перемещение», который можно использовать в стандартных потоковых диаграммах дискретно-событийных моделей. В качестве примера и прототипа для расширения нами был выбран «неделимый» компонент *moveTo* библиотеки моделирования процессов среды AnyLogic.

Во внутренней структуре разработанного компонента *nSTMmove* детально описывается поведение в ходе движения произвольного внешнего динамического агента от точки отправления в точку назначения в нестационарной окружающей среде по составляющим участкам, то есть с варьированием как скорости, так и траектории движения в зависимости от складывающейся обстановки. Реализованная внутри этого компонента логика встроенной маршрутизации заявок (внутренняя процессная диаграмма) приведена на рисунке 1.

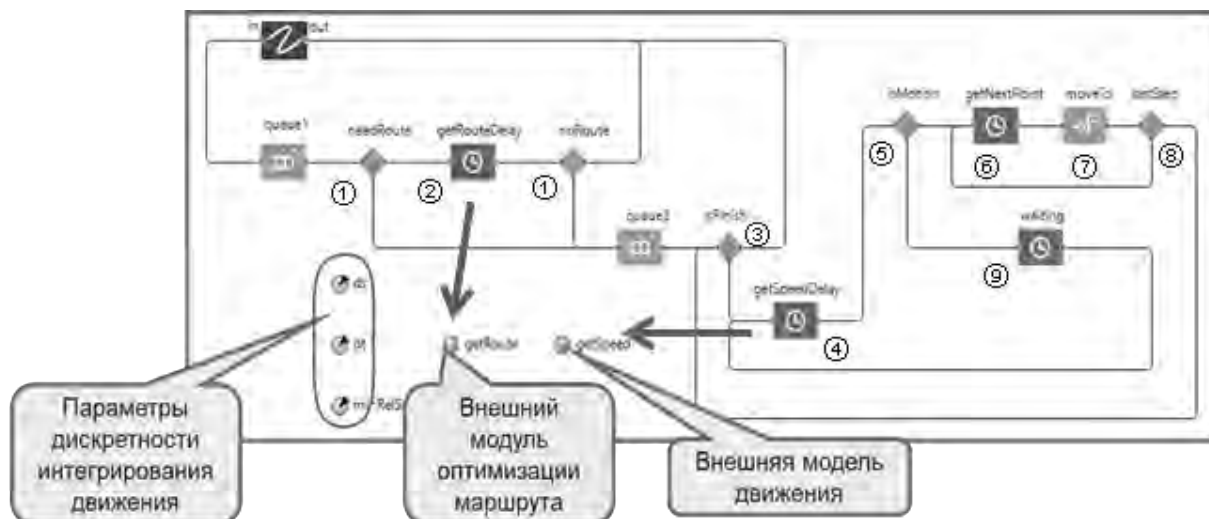


Рис. 1. – Внутренняя процессная диаграмма компонента *nSTMmove*, описывающая неравномерное движение агента в нестационарной среде

На вход данной процессной диаграммы поступает заявка, которая образуется во внешнем модуле, т.е. в головной процессной диаграмме. В заявке должны быть указаны точка отправления, точка назначения и идентификатор транспортного средства (судна), которым осуществляется данный рейс. В качестве внешних настроечных параметров компонента *nSTMmove* (т.е. статических членов его программного интерфейса) задаются характеристики внутреннего процесса интегрирования движения (временная и пространственная дискретизация), а также минимальный порог текущей скорости, который соответствует невозможности дальнейшего передвижения. Все поступающие на вход (порт *in*) заявки формально выстраиваются в очередь неограниченной вместимости. После этого в условии ① (здесь и далее рисунок 1) происходит проверка наличия в описании данной заявки предопределённой конкретной траектории, по которой должно происходить движение. Тем самым учитывается возможность явного указания траектории во внешнем вызывающем модуле. При отсутствии траектории, в блоке задержки ② происходит вызов функции *getRoute()*, в коде исполнения которой можно вставить обращение к произвольному внешнему модулю маршрутизации. Так, например, для простейшего случая по умолчанию в качестве данного модуля можно использовать элементарный маршрутизатор, который возвращает прямолинейный маршрут от точки старта к точке финиша. Однако этот метод может быть произвольным образом переопределен. Например, можно воспользоваться встроенными в платформу AnyLogic алгоритмами вызова внешних провайдеров маршрутов для сети автомобильных или железнодорожных путей или использовать внешние интеллектуальные алгоритмы ледового роутинга судов и караванов по электронным картам текущих и прогнозируемых условий окружающей среды (Торај et al., 2019). После попытки расчета маршрута следования вновь производится проверка его наличия у заявки. В случае, если маршрут так и не удалось получить, заявка сразу направляется в выходной порт компонента, в противном случае осуществляется переход к моделированию движения.

Моделирование движения, в свою очередь, заключается в последовательном выполнении логического цикла операций. Прежде всего, осуществляется проверка достижения точки назначения в условии ③. Если результат позитивен, то прохождение заявки по внутренней процессной диаграмме компонента *nSTMmove* прекращается, и она отправляется в выходной порт *out*. В противном случае осуществляется переход к

внутреннему циклу последовательных шагов интегрирования движения. Первая операция в данном цикле – вычисление достижимой скорости движения ТС в текущей точке времени и пространства с учетом сложившихся условий окружающей среды (4). Для этого осуществляется вызов функции *getSpeed()*, внутренняя реализация которой может быть произвольной. В качестве базового подхода здесь предполагается обращение к внешнему модулю расчета скорости и расхода топлива (модели ходкости судна).

Полученная величина скорости проверяется на превышение порога остановки (5). В случае, если рассчитанная скорость оказалась меньше критической, заявка переходит в режим ожидания (9), в котором находится период времени, равный текущему значению параметра временной дискретности интегрирования dt , после чего происходит повторный запрос скорости. Этот механизм позволяет отразить в подмодели движения факт «застревания» модельных судов в ходе движения при столкновении с критическим сочетанием погодных или ледовых условий. В случае же получения физического значения скорости происходит расчет следующей точки модельной траектории при прямолинейном движении (6). При этом одновременно рассматриваются несколько ограничений на длину очередного участка движения: не превышение дискретности интегрирования по времени и пространству, а также не выход за пределы текущего прямолинейного сегмента в заданном маршруте. После определения следующей точки происходит физическое перемещение объекта к ней. Перемещение происходит по прямой с определенной ранее скоростью, для этого используется стандартный компонент *moveTo* (7) процессной диаграммы AnyLogic. Шаги (6) и (7) повторяются до тех пор, пока не будет выбрана вся длина текущего шага интегрирования. Проверка этого условия выполняется в компоненте (8).

Использование данного компонента в головной процессной диаграмме модели, в которой заявки представляют собой рейсы транспортных судов, а ресурсы, например, привлекаемые для сопровождения ледоколы, позволяет реализовать контролируемое перемещение модельного объекта в нестационарной геоинформационной среде по произвольной траектории с постоянной коррекцией скорости путем обращения к расчетной модели движения. Заметим, что незначительная модификация разработанного компонента (более широкий контур обратной связи при достижении конца очередного прямолинейного участка) позволяет также осуществлять процедуру оперативной коррекции оставшегося маршрута, то есть возможность включения в логику модели процедуры динамически уточняющегося роутинга судов и караванов во льдах.

Результаты и обсуждение. Разработанный логический компонент неравномерного движения имеет статические и динамические составляющие программного интерфейса, которыми определяется его взаимодействие с головной моделью или процессной диаграммой. Статические параметры носят настроечный характер и определяют дискретность внутреннего процесса интегрирования движения. Понятно, что чем меньше выбраны величины ds и dt , тем более детально и точно описывается процесс перемещения, но, соответственно, тем большим количеством плановых элементарных событий будет представлен этот процесс в динамическом движении имитационной модели, то есть тем дольше будет происходить расчёт. В качестве динамических параметров (членов-функций) компонента выступают вынесенные наружу реализации методов *getRoute()* (динамическое определение рекомендуемой траектории перемещения на оставшемся участке маршрута) и *getSpeed()* (расчет скорости движения на каждом элементарном прямолинейном участке). При этом можно предложить два принципиальных подхода к имплементации соответствующих методов.

В первом случае прямо в ходе исполнения имитационной модели происходит обращение к соответствующим внешним расчётным процедурам, то есть модуль ледового роутинга или модель движения судна вызываются непосредственно из кода имитационной модели. Например, подобный подход использовался авторами при разработке имитационной модели снабжения платформ разведочного бурения в акватории Обской и Тазовской губ. Характерными особенностями изучаемой транспортной системы в данном случае выступали: а) сравнительно узкий пространственный масштаб региона интереса; б) многообразие и разнотипность транспортных средств снабжения (водоизмещающие суда ледового класса, суда на воздушной подушке, вездеходы, вертолёты); в) высокая динамичность внешних условий как в пространстве, так и во времени; г) существенное влияние внешних условий на скорость движения транспортных средств. Поэтому в данном случае использование разработанного компонента производилось путём обращения к встроенным моделям движения собственной разработки (Tarovik et al, 2017), а параметры дискретизации интегрирования движения выбирались достаточно малыми (например, для того, чтобы не «проскочить» узкий участок торошёного льда, что не исключено при выбранной явной схеме интегрирования). К плюсам подобного решения следует отнести то, что при этом вся сложная логика модели оказывается инкапсулирована внутри используемого универсального компонента (на рис. 2 показано какой простой вид при этом приобретает головная процессная диаграмма), а к недостаткам – существенное замедление эффективности и быстродействия модельных расчётов.

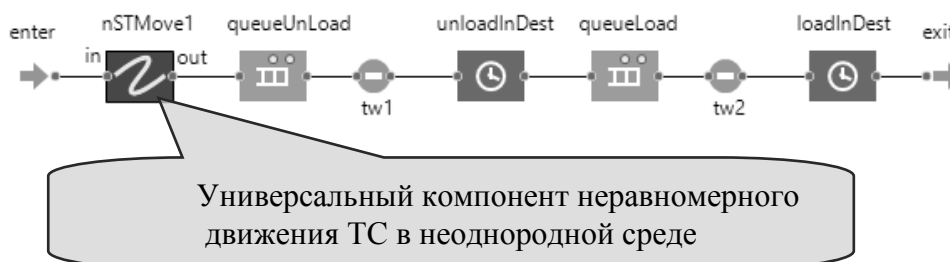


Рис. 2 – Использование универсального компонента неравномерного перемещения в процессной диаграмме имитационной модели системы снабжения буровых платформ

Последнее обстоятельство приводит к тому, что более подходящим для исследования систем крупного пространственного масштаба является другой способ использования описанного универсального компонента. Именно он был применён авторами при разработке модели транспортной системы Северного морского транзитного коридора – высокоширотной контейнерной линии на основе контейнеровозов высокого ледового класса, которая рассматривается в качестве перспективной альтернативы южному маршруту евро-азиатского транзита грузов через Суэцкий канал. В этом случае масштаб региона интереса включает в себя всю акваторию Северного морского пути (СМП), поэтому прямое обращение к внешним сервисам оптимальной маршрутизации или детальным моделям движения в каждой точке каждого рейса каждого модельного судна непосредственно в ходе прогона модели оказывается практически неосуществимым. Вместо этого в рамках упомянутого проекта опорные траектории оптимальных маршрутов движения арктических контейнеровозов по рёбрам предопределённого топологического графа переходов по СМП были рассчитаны заранее и сохранены в виде специальной базы данных оптимальных маршрутов. Для осуществления этих массовых расчётов для всех внедрённых в систему архивных ледовых карт была написана специальная

программная утилита. Общее число задач маршрутизации, которые следовало решить для целей проекта, составило 7 вариантов судов × 9 тестовых сезонов × 52 карты в сезоне × 24 ребра × 2 режима (с ледоколом и без ледокола), что даёт в результате более полутора сотен тысяч элементарных расчётов. Для вычисления такого количества маршрутов был использован выделенный вычислительный сервер с 16 процессорами, на которых в параллельном режиме было запущено 16 экземпляров разработанной сервисной утилиты каждый со своим списком кортежей вычисляемых характеристик. Суммарное время непрерывных расчётов составило около одной недели. Результатом этого процесса явилась заполненная реляционная база данных оптимальных маршрутов. Именно она использовалась в ходе прогона имитационной модели при исполнении агентами – транспортными судами-контейнеровозами заданий на перевозку. То есть при данном подходе семантика функций универсального компонента *getRoute()* и *getSpeed()* сводится просто к чтению массива маршрутных точек и характерных параметров движения для каждой из них из внешней предрассчитанной базы данных. Как результат, визуальное перемещение судов-агентов осуществляется в геоинформационной среде согласно предрассчитанным оптимальным маршрутам движения в данный момент модельного времени, в соответствии с текущей ледовой картой, и отдельно для каждого ребра глобального графа перевозок (рис. 3).

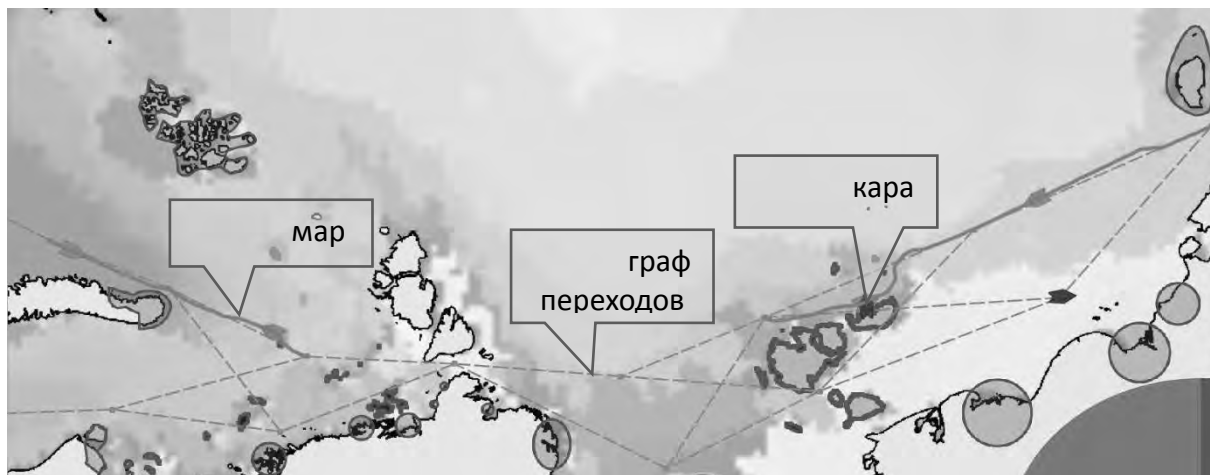


Рис. 3 – Движение судов-агентов по предрассчитанным оптимальным маршрутам в ходе исполнения имитационной модели транспортной системы СМТК

Литература

1. **Топаж А.Г., Таровик О.В., Бакhareв А.А., Кондратенко А.А.** Optimal ice routing of a ship with icebreaker assistance // Applied Ocean Research, 2019, (86), pp. 177–187.
2. **Таровик О.В., Топаж А.Г., Бакhareв А.А., Косоротов А.В., Крестьянтсев А.В., Кондратенко А.А.** Multidisciplinary approach to design and analysis of arctic marine transport systems // Proceedings of the 36th International Conference on Ocean, Offshore & Arctic Engineering (OMAЕ 2017). – 2017.