

Алгоритм «Переправа» (Алгоритм для программирования задач многоагентного моделирования с использованием методов дискретно-событийного моделирования)

В.И. Тихвинский, В.В. Холмогоров

Аннотация – Рассмотренный в данной статье алгоритм «Переправа» использовался в учебном процессе при преподавании дисциплин «Имитационное моделирование» и «Многоагентное моделирование». Подобную задачу студенты сначала выполняли в среде AnyLogic, но в AnyLogic практически не осваиваются учащимися навыки программирования. Именно поэтому данная задача решается ими как в среде AnyLogic, так и в определенной среде соответствующего языка программирования по выбору. Для решения данной задачи хорошо использовать визуальную среду программирования. Рассматриваемой среде можно использовать элемент управления таблицу (см. рисунок 1). Но как показал наш преподавательский опыт, студенты могли обходиться и без данного элемента. Некоторые даже решали эту задачу, когда монитор компьютера был не в графическом, а в текстовом режиме. Такое решение, конечно менее выразительно, но вполне допустимо. Так что практика показала, что данный алгоритм может быть использован учащимися, которые имеют неодинаковый уровень в программировании, и имеют опыт программирования в разных языках. При выполнении заданий, студенты могут несколько видоизменить выполнения в нем событий, например, люди могут приходить на пристань или переходить на плот не по одному, а строем. Но в целом он должен выполнять поставленную задачу.

Алгоритм может также при учебном процессе быть видоизменен и усложнен, например, плот в нем может быть заменен на кабину лифта, который будет двигаться по этажам, перевоза людей с одного этажа на другой, этажей может быть не только два, но при усложнении алгоритма суть его останется прежней.

Ключевые слова – имитационное моделирование, многоагентное моделирование, алгоритм «Переправа», программирование, преподавание.

ВВЕДЕНИЕ

Модель являет собой образ реального объекта, который представляется различным способом. Модель используется в качестве заменителя реального объекта, процесса или явления. Сам же объект, явление, процесс называется оригиналом созданной модели [1].

Дискретно-событийное моделирование это метод, при котором происходит отход от непрерывной цепочки событий природы, и при моделировании какой-либо системы рассматриваются только определенные события, например: «обработка заказа», «движение с грузом», «разгрузка» и другие. Данный вид моделирования основан Джеффри Гордоном в 1960-х годах [2].

Определим самые основные понятия дискретно-событийного моделирования.

Система – представляет собой множества объектов, иными словами агентов, которые выполняют какое-либо действие. В качестве агентов могут рассматриваться люди, машины и др., которые взаимодействуют между собой. Их взаимодействие приводит к выполнению определенной задачи.

Модель – есть условное представление системы, построенное на логических или математических отношениях, которые описывают систему в целом [3].

Событие – это возникающее изменение состояния системы, при котором агент или агенты производят определенное действие [3].

Действие – есть определенный промежуток времени, на котором агент или агенты выполняют часть поставленной им задачи [4].

Задержка – это приостановка выполнения какого-либо действия на определенный промежуток времени [4].

Система, в которой несколько агентов могут взаимодействовать между собой и решать поставленную задачу называется многоагентной. В многоагентных системах подзадачи распределены между различными агентами и/или группой однотипных агентов. В данной статье группой агентов являются люди, изображаемые в виде смайликов (см. рисунок 1). Их подзадачи: прийти на пристань, перейти на плот, сойти с плота, уйти с пристани. Эти подзадачи будут выполнены при соответствующих событиях: Люди на пристань, Люди на плот, Люди с плота, Люди с пристани. Одиночным агентом в статье является плот, подзадачи которого следующие: переправить людей на противоположный берег, вернуться пустым на исходный берег. Эти подзадачи выполняются при соответствующих событиях: Движение влево, Движение вправо. Теперь перейдем к постановке задачи.

Статья получена 11.06.2022 г.

В.И. Тихвинский, МИРЭА (e-mail: tvitaly1@yandex.ru).

В.В. Холмогоров, МИРЭА (e-mail: Hv13@mail.ru).

ПОСТАНОВКА ЗАДАЧИ

Имеются объекты плот, водное пространство, правая и левая пристань, и люди, переправляющиеся на плоту с правой пристани на левую. Объект плот обозначен соответствующей надписью и серой заливкой, водное пространство обозначено синей заливкой, а пристани черной. Люди обозначены символом смайлика ☺.

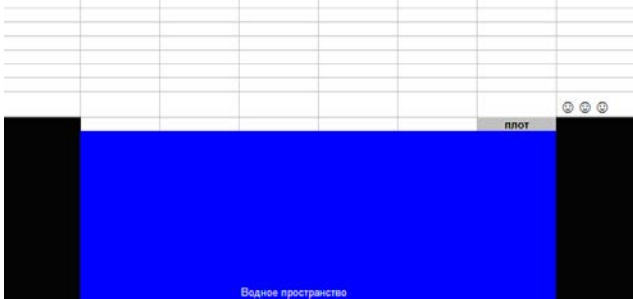


Рисунок 1 – Люди пришли на пристань для переправы

Необходимо, что бы на правую пристань пришло случайное число людей, от одного до пяти человек. Затем, пришедшие на пристань люди перешли на плот. Помимо сказанного, необходимо, чтобы плот переправил людей на левую пристань, что бы люди сошли с плота на пристань и ушли с левой пристани. И после всего, плот должен вернуться на правую пристань. Данный список действий должен повториться заданное число раз [5]. О событиях, необходимых для решения данной задачи, было сказано в предыдущем разделе этой статьи в последнем абзаце. Поэтому перейдем к основному алгоритму.

ОСНОВНОЙ АЛГОРИТМ

Блок-схема основного алгоритма показана нами на рисунке 2. Вначале вводится значение переменной k , которая задает количество итераций повторения группы событий. Условие повторения событий определяет соответствующее логическое выражение $k > 0$. Блоки, обозначающие события, залиты светлым желтым цветом. Алгоритмы событий будут рассмотрены в следующих разделах данной статьи. Как явствует из блок-схемы, после выполнения группы событий, значение переменной k уменьшается на единицу, а затем вновь проверяется условие повторение группы событий. Если оно истинное, вновь происходит вход в тело цикла, в противном же случае демонстрация модели завершается [5].

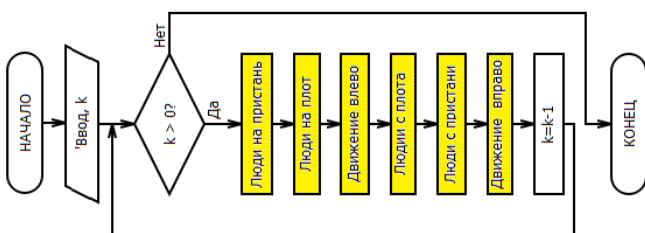


Рисунок 2 – Блок-схема основного алгоритма

СОБЫТИЕ «ЛЮДИ НА ПРИСТАНЬ»

Блок-схема алгоритма события «Люди на пристань» представлена на рисунке 3. Сначала переменной k присваивается случайное число от 1 до 5. Затем проверяется условия входа в цикл $k > 0$. Если оно истинно, добавляется символ ☺ в ячейку над правой пристанью, уменьшается значение k и происходит приостановка алгоритма через задержку. Приостановка необходима, что бы было видно, как люди приходят на пристань постепенно, по одному. Цикл повторяется, пока значение переменной k не станет равным нулю.

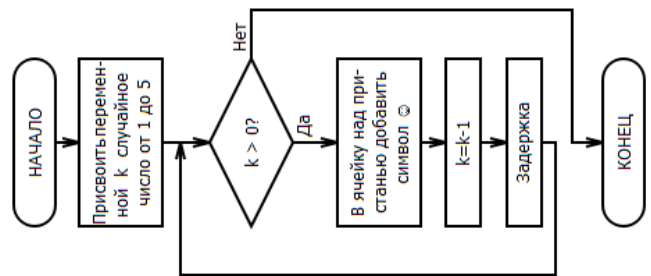


Рисунок 3 – Приход людей на правую пристань

СОБЫТИЕ «ЛЮДИ НА ПЛОТ»

Блок-схема алгоритма события «Люди на плот» представлена на рисунке 4. Сначала проверяется наличие текста в ячейке над пристанью, если ячейка не пуста, значит и пристань не пуста (см. рисунок 1). Далее происходит вход в цикл и один символ ☺ удаляется из ячейки над пристанью, а затем добавляется символ ☺ в ячейку над плотом. После происходит приостановка алгоритма, что бы было видно, как люди по одному переходят с пристани на плот. Далее цикл повторяется до тех пор, пока пристань не станет свободной, иными словами все люди с пристани перейдут на плот [5].

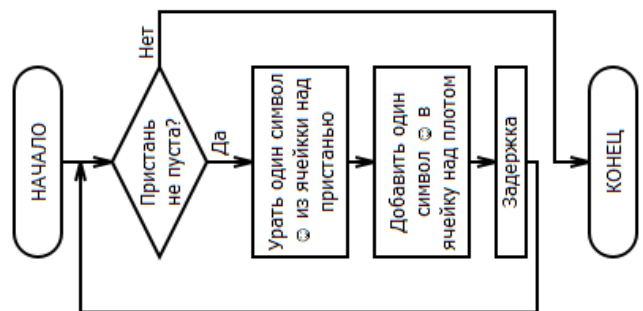


Рисунок 4 – Переход людей на плот с правой пристани

СОБЫТИЯ «ДВИЖЕНИЕ ВЛЕВО» И «ЛЮДИ С ПЛОТА»

Блок-схема алгоритма события «Движение влево» представлена на рисунке 5. Сначала проверяется условие «причалил ли плот (к левой пристани)». Это определяется наличием черной заливки слева от плота

(см. рисунок 1). Если плот не причалил, согласно блок-схеме производится перемещение содержимого ячейки с плотом и содержимого ячейки над ним в соседние ячейки слева. Затем происходит приостановка алгоритма за счет задержки. Затем горизонтальная координата ячейки j , в которой находился плот, уменьшается на единицу. Затем вновь проверяется условие «причалил ли плот». Далее цикл повторяется до тех пор, пока плот с людьми не достигнет левой пристани [5].

Блок-схема алгоритма события «Люди с плота» представлена на рисунке 6. Сначала проверяется наличие текста в ячейке над плотом, если ячейка не пуста, значит и плот не пуст. Далее происходит вход в цикл и один символ ☹ удаляется из ячейки над плотом, а затем добавляется символ ☹ в ячейку над левой пристанью (см. рисунок 1). После происходит приостановка алгоритма за счет задержки. Далее цикл повторяется до тех пор, пока плот не станет свободным, иными словами все люди с плота перейдут на пристань [5].

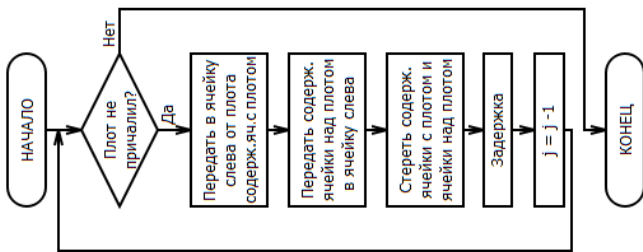


Рисунок 5 – Движение плота с людьми к левой пристани

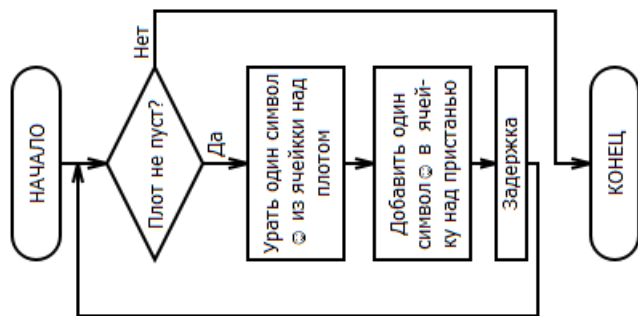


Рисунок 6 – Переход людей с плота на левую пристань

СОБЫТИЕ ЛЮДИ С ПРИСТАНИ И СОБЫТИЕ ДВИЖЕНИЕ ВПРАВО

Блок-схема алгоритма события «Движение вправо» представлена на рисунке 8. Сначала проверяется условие «причалил ли плот (к правой пристани)». Это определяется наличием черной заливки справа от плота (см. рисунок 1). Если плот не причалил, согласно блок-схеме производится перемещение содержимого ячейки с плотом в соседнюю ячейку справа. Затем происходит приостановка алгоритма за счет задержки. Затем горизонтальная координата ячейки j , в которой находился плот, увеличивается на единицу. Затем вновь

проверяется условие «причалил ли плот». Далее цикл повторяется до тех пор, пока пустой плот не достигнет правой пристани [5].

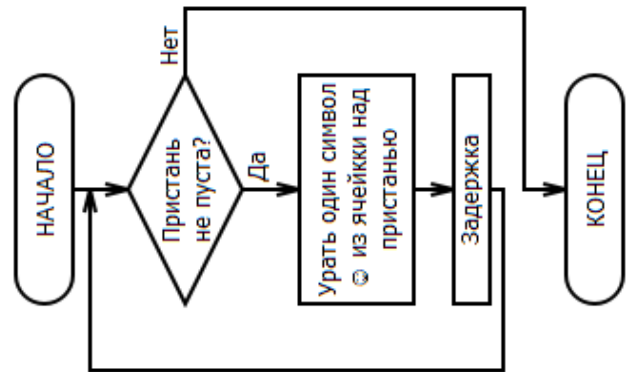


Рисунок 7 – Переход людей с плота на левую пристань

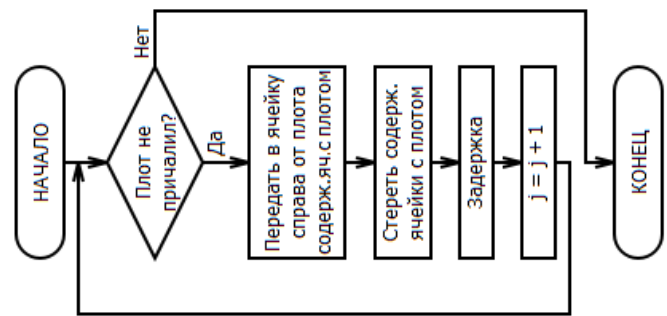


Рисунок 8 – Движение плота к правой пристани

ЗАКЛЮЧЕНИЕ

Код основного алгоритма пишется в специально предназначенной для этого основной подпрограмме или, если используется классический язык программирования, в основной программе. Коды алгоритмов событий пишутся в своих подпрограммах с определенными именами, события вызываются по имени из кода основной подпрограммы. Задержка также является подпрограммой. Она может быть организована циклом с большим количеством итераций. Для замедления работы цикла, в его тело записывают выполнение какой-либо математической операции, например, присвоение математического выражения числовой переменной, в котором используются неименованные числовые константы: $s=5*10/2$. В более сложно организованных задержках используются проверки по системному таймеру. Как явствует из блок-схем алгоритмов, событий подпрограмма задержки для данной задачи должна запускаться из кода подпрограммы соответствующего события.

БИБЛИОГРАФИЯ

- [1] Тихвинский В.И., Платонова О.В. Компьютерное моделирование, Методические указания: Практикум. / – М.: «МИРЭА – РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ» РТУ

- МИРЭА, 2020. – 66с. – 1 электрон. опт. диск (CD-ROM).
- [2] Кельтон В. Имитационное моделирование. Классика CS / В. Кельтон, А. Лоу. 3-е изд. – СПб.: Питер; К: Издательская группа BHV, 2004. – 847 с.
- [3] Павловский Ю. Н. Имитационное моделирование / Ю.Н. Павловский, Н.В. Белотелов, Ю.И. Бродский. – М.: Академия, 2008. – 235 с.
- [4] Емельянов, В. В. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО / В. В. Емельянов, С. И. Ясиновский – М.: «АНВИК», 1998. – 427 с.
- [5] Тихвинский В.И., Сорокин А.Б. Имитационное моделирование: практикум [Электронный ресурс]: Учебно-методическое пособие / Тихвинский В.И., Сорокин А.Б. – М.: Московский технологический университет (МИРЭА), 2018, – 161с.

Algorithm "Crossing" (Algorithm for programming tasks of multi-agent modeling using methods of discrete-event modeling)

V. I. Tikhvinsky, V.V. Kholmogorov

Abstract – The “Crossing” algorithm considered in this article was used in the educational process when teaching the disciplines “Simulation modeling” and “Multi-agent modeling”. Students first performed a similar task in the AnyLogic environment, but students practically do not master programming skills in AnyLogic. That is why they solve this problem both in the AnyLogic environment and in a certain environment of the corresponding programming language of their choice. To solve this problem, it is good to use a visual programming environment. The considered environment can use the table control (see Figure 1). But as our teaching experience has shown, students could do without this element. Some even solved this problem when the computer monitor was not in graphic mode, but in text mode. Such a solution, of course, is less expressive, but quite acceptable. So practice has shown that this algorithm can be used by students who have different levels in programming and have programming experience in different languages. When completing tasks, students can somewhat modify the execution of events in it, for example, people can come to the pier or go to the raft not one at a time, but in formation. But overall, it should do the job.

The algorithm can also be modified and complicated during the educational process, for example, the raft in it can be replaced by an elevator cabin that will move through the floors, transporting people from one floor to another, there can be not only two floors, but when the algorithm becomes more complicated, its essence will remain the same.

Key words - simulation modeling, multi-agent modeling, "Crossing" algorithm, programming, teaching.

REFERENCES

- [1] Tikhvinsky V.I., Platonova O.V. Computer modeling, Guidelines: Workshop. / - M.: "MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY" RTU MIREA, 2020. - 66s. – 1 electron. opt. disc (CD-ROM).
- [2] Kelton V. Simulation. Classics CS / V. Kelton, A. Lowe. 3rd ed. - St. Petersburg: Peter; K: BHV Publishing Group, 2004. - 847 p.
- [3] Pavlovsky Yu. N. Simulation / Yu.N. Pavlovsky, N.V. Belotelov, Yu.I. Brodsky. – M.: Academy, 2008. – 235 p.
- [4] Emelyanov, VV Introduction to intelligent simulation modeling of complex discrete systems and processes.

Language of the RDO / V. V. Emelyanov, S. I. Yasinovsky - M.: "ANVIK", 1998. - 427 p.

[5] Tikhvinsky V.I., Sorokin A.B. Simulation modeling: workshop [Electronic resource]: Educational and methodological manual / Tikhvinsky V.I., Sorokin A.B. – M.: Moscow Technological University (MIREA), 2018, – 161p.