

УДК 004.94

АЛЬТЕРНАТИВНЫЙ ПОДХОД К ПОСТРОЕНИЮ ИМИТАЦИОННЫХ МОДЕЛЕЙ НА ОСНОВЕ СИСТЕМЫ ЗАПРОСОВ И ЗАДАЧ

В.Е. Черненко, А.А. Малыханов, А.Д. Шуравин (Ульяновск)

Введение

Современные системы имитационного моделирования, включая AnyLogic[1] и Simio[2], часто предполагают создание моделей в режиме «Point&Click», где логика процесса описывается с помощью визуальных диаграмм процессов (Process Modeling).

Например, логика зоны приемки склада, изображенного на рис.1, может быть представлена в виде диаграммы на рис. 2. [3]

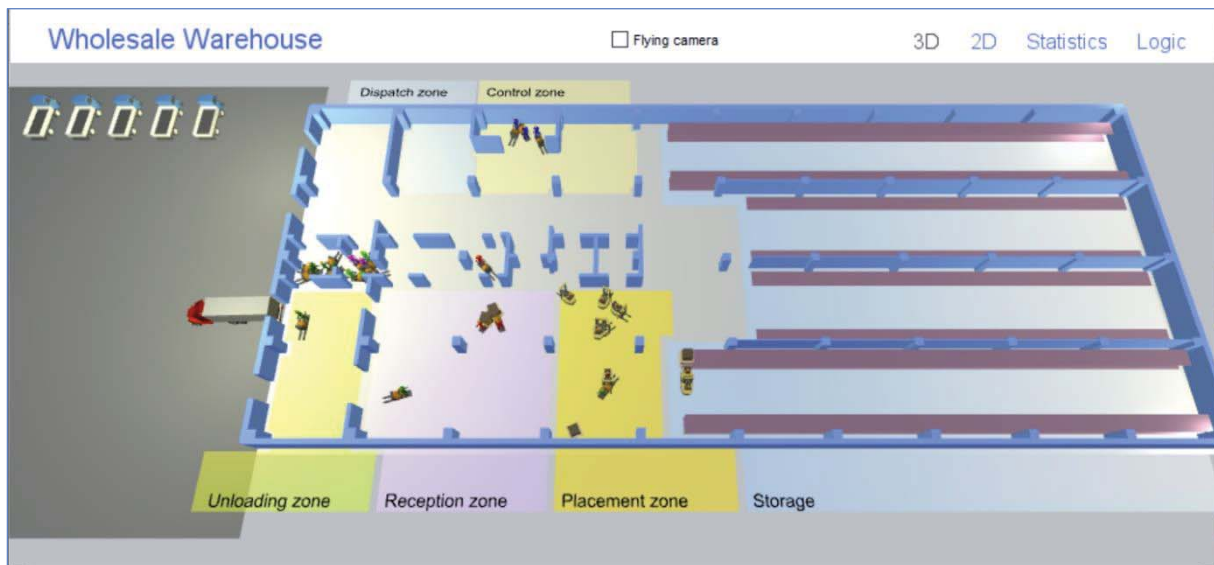


Рис.1. Модель в AnyLogic с зоной приемки с двумя воротами

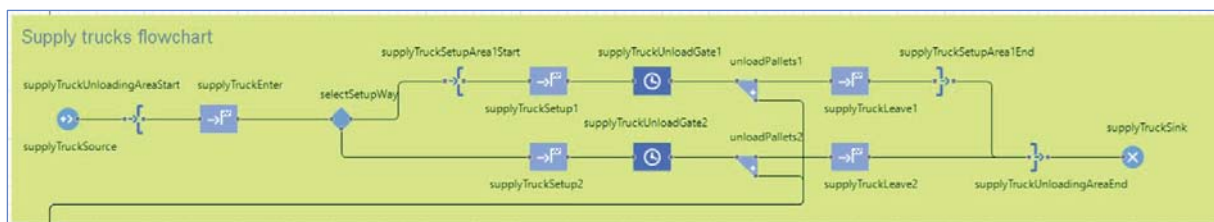


Рис.2. Диаграмма, описывающая логику работы зоны приемки

Хотя такой подход интуитивно понятен и существенно упрощает начальную разработку, он обладает следующими недостатками:

- сложность модификации: любое изменение логики работы требует перестройки диаграмм, что быстро приводит к их усложнению; на приведенном примере добавление складских ворот требует добавления как минимум двух элементов, соответствующих этим воротам на процессную диаграмму;
- связность логики и геометрии: геометрические элементы (например, количество ворот, расположение стеллажей) жестко встроены в логику процесса; для проверки гипотезы о работе склада с 3-я или 4-я входными воротами необходимо создавать отдельные модели или проводить сложную реконфигурацию;

- проблемы с тестированием: сложности в процессе изолированного тестирования отдельных компонентов логики модели, заданной с помощью процессной диаграммы. [4]

Агентный подход частично решает эти проблемы, вынося логику в поведение активных агентов (например, погрузчиков или грузчиков). Логика агента часто описывается с помощью диаграмм состояний (Statechart), что делает модель более гибкой (рис.3). [5]

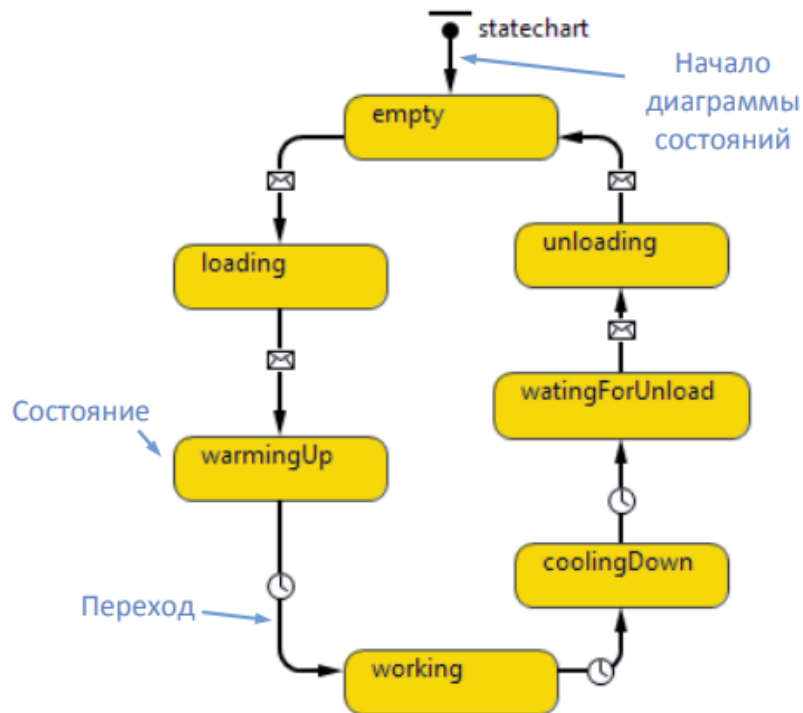


Рис.3. Диаграмма состояний

Однако при моделировании сложных систем диаграмма состояний агента может стать чрезмерно сложной [6]. Решением является разделение логики на несколько диаграмм состояний. Но в этом случае придется столкнуться с их пересечением и возникновением неявных зависимостей (например, как обработать поломку оборудования, которое в данный момент находится в состоянии «Перемещение» или «Загрузка»).

Предлагаемый подход

Для решения описанной проблемы предлагается подход, который заключается в выделении двух ключевых абстракций: запрос (Request) и задача (Task).

Запросы

Запрос – это описание действия, которое должно быть выполнено в моделируемой системе для достижения цели, например, «разгрузка грузовика в зоне приемки», «бурение скважин в горной выработке», «выполнение фрезерной операции» и т.д. Запросы запускают работу имитационной модели: если нет запросов – ничего не происходит.

Например, для модели склада паллетного хранения типичными запросами являются:

- запрос на разгрузку грузовика в зону промежуточного хранения;

- запрос на перемещение паллет из зоны промежуточного хранения в зону основного хранения;
- запрос на подбор паллеты со стеллажей;
- запрос на погрузку паллет из зоны подбора в грузовик.

Запросы могут генерироваться по некоторым правилам. Например, согласно закону распределения, а также появляться в процессе выполнения других запросов, например, завершение разгрузки грузовика может создать один или несколько запросов на перемещение разгруженных паллет в зону основного хранения.

Запрос характеризуется атрибутами, такими как время создания запроса, время начала работы над ним, процент выполнения запроса и т.д. Ключевая особенность запроса – один запрос может быть выполнен несколькими разными агентами, возможно, с прерываниями. Например, запрос на подбор 12 паллет может быть начат грузчиком с рохлей, а завершён погрузчиком. А также такой запрос может выполняться одновременно более чем одним агентом.

Задачи

Если запрос определяет «что» нужно сделать, то задача определяет «как» это сделать конкретным агентом. Для одного запроса может существовать несколько возможных задач. Например, выполнение анкерной крепи в горной выработке может выполняться специальным оборудованием (анкероустановщиком) или с помощью бурового станка и ручной установки анкеров; транспортировка груза из пункта А в пункт Б может выполняться по разным маршрутам; фрезерная операция сможет выполняться на собственном станке или силами подрядчика за фиксированное время и т.д.

Задача содержит детальную информацию о том, как будет выполняться запрос, и характеризуется специфическим набором атрибутов, например, продолжительность выполнения операции данной единицей оборудования, выбранный маршрут и стоимость выполнения транспортировки по нему.

Задача, а не агент, содержит в себе диаграмму состояний, которая описывает последовательность простых действий агента для выполнения запроса: «поднять паллету», «переместиться в точку Х», «опустить паллету». А агент содержит в себе логику выполнения этих простых действий. При этом одно такое простое действие может быть частью процесса выполнения задач различных типов. Правильность выполнения простых действий может быть отдельно протестирована с помощью набора Unit-тестов [4].

Таким образом, агент становится универсальным исполнителем, а его сложное поведение определяется задачами, которые он может выполнять. Задача является объектом, контролирующим агента на протяжении своего исполнения.

Вспомогательные задачи

Оборудование (агенты) имеют собственные потребности: зарядка батареи, заправка топливом, техническое обслуживание и т.д. Эти потребности также моделируются как отдельные задачи, например, «задача зарядки батареи». Диаграмма состояний этих задач описывает последовательность действий для удовлетворения этих потребностей. Это создает единый механизм для описания всей активности в системе [7].

Ядро системы: диспетчеризация и учет расписаний

Диспетчеризация выполняет 2 основные функции:

- определяет порядок выполнения заявок с учетом приоритетов;

- формирует задачи со всей необходимой информацией для их исполнения.

В общем случае диспетчеризация должна выполняться в моменты времени, когда меняется состояние моделируемой системы:

1. появляется новая заявка;
2. становится доступным оборудование.

При появлении новой заявки диспетчер выполняет поиск подходящего свободного оборудования для ее выполнения. Если оборудование найдено, создается задача со всеми необходимыми параметрами. Если подходящего свободного оборудования нет, то заявка помещается в очередь заявок.

Каждый раз, когда оборудование завершает текущую задачу, в том числе вспомогательную, оно обращается к диспетчеру. Диспетчер проверяет состояние оборудования (остаток топлива, время до конца смены и т.д.) и с учетом этой информации выбирает следующую заявку для выполнения и формирует задачу. Если внутренняя потребность оборудования критична, создается соответствующая вспомогательная задача. Если агент готов к работе, диспетчер выбирает следующую заявку из очереди заявок на основе приоритетов и создает основную задачу.

Диспетчеризация решает вопрос учета расписаний, так как пересменка или простой – это тоже задачи.

В случае необходимости сложной диспетчеризации, например, когда для выполнения одной заявки требуется более одного типа оборудования, диспетчеризация может выполняться в начале смен, когда большинство оборудования свободно.

Правильность выполнения диспетчеризации может быть протестирована в отрыве от характеристик оборудования и от запуска всей имитационной модели [4].

Преимущества подхода

- гибкость и модульность: логика выполнения задач полностью отделена от логики диспетчеризации; геометрия системы влияет только на параметры задач (координаты точек), но не на их последовательность;
- тестируемость: логику отдельной задачи (например, алгоритм размещения паллеты на стеллаже) можно протестировать изолированно, за счет проверки правильности работы ИМ с одной созданной в заданное время задачей; аналогично можно тестировать алгоритм диспетчеризации на упрощенной модели без полнофункциональной имитации процессов;
- чистота архитектуры: точки принятия решений (диспетчер) четко отделены от кода, реализующего выполнение задач; это упрощает совместную разработку и модификацию модели;
- адекватность реальным процессам: подход легко позволяет моделировать сложные сценарии: прерывание задач, выполнение одного запроса несколькими агентами, учет внутренних потребностей оборудования.

Заключение и дальнейшие направления

Описанный подход к построению имитационных моделей на основе системы запросов и задач не имеет недостатков классических методов. Он был успешно применен в нескольких коммерческих проектах по моделированию сложных систем в логистике, производстве и горной добыче.

В рамках дальнейших публикаций планируется детализировать:

- использование алгоритмов комбинаторной оптимизации и библиотек линейного программирования для планирования и оптимизации порядка выполнения запросов в ходе моделирования;

- выделение стандартных блоков логики для моделирования типичных аспектов поведения агентов: перерывы по расписанию, пополнение расходных материалов и т.п.

Литература

1. Официальный сайт компании AnyLogic Company – Режим доступа <http://www.anylogic.ru/> (дата обращения 20.09.2025).
2. Официальный сайт компании SIMIO LLC (США) – Режим доступа <http://www.simio.com/index.php> (дата обращения 20.09.2025).
3. **Никифоров В. В.** Логистика. Транспорт и склад в цепи поставок. М.: ГроссМедиа, 2008.
4. **Малыханов, А. А.** Тестирование имитационных моделей / А. А. Малыханов, В. Е. Черненко // Десятая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД-2021) : Труды конференции (электронное издание), Санкт-Петербург, 20–22 октября 2021 года / Редакторы Плотников А.М., Долматов М.А., Смирнова Е.П.. – Санкт-Петербург: АО «Центр технологии судостроения и судоремонта», 2021. – С. 30-39. – EDN TMZXVS.
5. **Сычев, А.В.** AnyLogic за три дня [Электронный ресурс] / А.В. Сычев. — Электрон. дан. – [Б. м.] : AnyLogic, 2021. — 240 с. – Режим доступа: [https://www.anylogic.com/upload/al-in-3-days/anylogic-in-three_days\(rus\).pdf](https://www.anylogic.com/upload/al-in-3-days/anylogic-in-three_days(rus).pdf) (дата обращения: 20.09.2025).
6. **Бусленко, Н.П.** Моделирование сложных систем. – М.: Наука, 1978.
7. **Law, A. M.** Simulation Modeling and Analysis. – 5th ed. – McGraw-Hill, 2014.