

УДК 681.3

В.Д. Левчук

БАЗОВАЯ СХЕМА ФОРМАЛИЗАЦИИ СИСТЕМЫ МОДЕЛИРОВАНИЯ МІСІС4

Рассматривается базовая схема формализации, лежащая в основе системы моделирования МІСІС4, в которой реализован транзактный способ имитации. Полная функциональность МІСІС4 обеспечивается программируемым интерфейсом, который по своей сути близок к понятию "процесс". Определены базовые элементы и их свойства, статическая иерархическая структура и потоковая граф-схема имитационной модели, особенности реализации управляющей программы моделирования, графическое отображение механизмов обслуживания. Приводится методика формализации сложной системы согласно предложенной базовой схеме.

Введение

Новая платфо́рмо-независимая версия системы имитационного моделирования МІСІС4 (далее просто МІСІС4) является развитием версии 2.0 [1], реализованной в DOS-среде. В основе МІСІС4 лежит более мощная базовая схема формализации (БСФ), под которой понимается совокупность понятий, используемых для построения формального описания моделируемой системы и непосредственно представленных в языке моделирования [2].

В практике моделирования существует ряд способов формализации функционирования исследуемой системы [1-3]. Одним из них является *транзактный способ* [4, 5], когда объект моделирования представляется как сеть массового обслуживания (СМО). Для СМО характерны следующие статические элементы:

- источники заявок на обслуживание – транзактов;
- обслуживаемые устройства (приборы);
- очереди для сохранения транзактов перед занятыми устройствами;
- поглотители заявок;
- множество дополнительных элементов (блоков), управляющих движением транзактов и модифицирующих состояние как обслуживаю-

щих устройств, так и самих дополнительных элементов.

При структуризации сложной системы (СС) с помощью БСФ МІСІС4 также предполагается использование транзактной схемы. Однако число базовых элементов сведено к минимуму в отличие от других средств моделирования данного типа. При этом полная функциональность МІСІС4 (в смысле возможности отображения взаимодействия подсистем и элементов практически любой СС на высоком уровне детализации) реализуется за счет того, что обслуживание транзакта на устройстве обеспечивается программируемым интерфейсом, который по своей сути близок к понятию "процесс" [1, 3, 6].

Языки имитационного моделирования включают в себя конструкции, соответствующие определенной БСФ, и по синтаксической основе подразделяются на три категории [2]: с собственным синтаксисом, расширяющие универсальный язык программирования или вложенные в него. Последняя группа языков по форме является библиотекой функций, объектов, шаблонов и т.д. Язык моделирования МІСІС4 также принадлежит данной группе и его интерфейс реализован системным модулем на объектно-ориентированном языке программирования С++ [7]. Сис-

темный модуль можно оттранслировать отдельно от информационного модуля, описывающего взаимодействие составных элементов СС, а затем включать в проект любой имитационной модели (ИМ).

Таким образом, для написания программ ИМ с помощью MICIS4 разработчик модели должен владеть классическим универсальным языком программирования С++, а также изучить программный интерфейс, являющийся отображением БСФ MICIS4. Ее основные положения раскрываются в данной статье.

1. Базовые элементы MICIS4

Построение формальной модели рекомендуется начинать с декомпозиции СС на подсистемы и элементы [1, 8]. Разработчик ИМ средствами MICIS4 рассматривает объект моделирования как совокупность следующих базовых элементов: генераторов внешних событий (транзактов, в частности), обслуживающих устройств и транзактов (заявок на обслуживание).

Устройства предназначены для обслуживания транзактов. Процесс обслуживания заключается в программной имитации функциональных действий, совершаемых взаимодействующими элементами объекта моделирования. Обслуживание начинается с приходом транзакта на устройство и завершается после его ухода, т.е. без транзактов подпрограммы устройств не получают управление. Транзакт после обслуживания на исходном устройстве может переместиться на следующее (целевое) устройство, быть уничтожен системой моделирования, ожидать начала обслуживания или привести к аварийному окончанию эксперимента из-за невозможности его обслуживания целевым устройством.

Элемент модели, самостоятельно порождающий произвольные события, называется генератором.

Он позволяет отображать действие внешней среды как путем создания потока транзактов на обслуживающие, так и изменением состояния элементов модели. Кроме того, генераторы можно использовать для внутренней синхронизации элементов или достижения специальных целей при программировании ИМ (например, отладки, верификации или мониторинга состояния ИМ).

Множество необходимых для эксперимента с ИМ устройств и генераторов требуется задавать заранее. С другой стороны, их можно уничтожить только по окончании эксперимента с ИМ. Транзакты, наоборот, первоначально отсутствуют, а затем добавляются и уничтожаются в прогоне ИМ в произвольном порядке и количестве (точнее, ограниченном только имеющимся в наличии объемом оперативной памяти). В этом смысле транзакты являются динамическими элементами модели, а устройства и генераторы — статическими элементами.

Как известно, при дискретно-событийном моделировании состояния сложной системы изменяются в дискретные моменты модельного времени, т.е. в моменты возникновения событий [1, 2]. Изменение состояния модели реализуется путем вызова из управляющей программы моделирования (УПМ) обработчика события, который называется активностью и представляет собой функцию или процедуру языка моделирования.

В MICIS4 функционирование генератора в ИМ представляет собой множество активностей, объединенных в виде связного ориентированного графа, в котором дуги показывают возможные пути возникновения событий, обрабатываемых активностями — вершинами. Никаких ограничений на структуру данного графа не накладывается, т.е. допускаются вместе с линей-

ными участками условные переходы и циклы. После выполнения любой активности генератора управление передается в УПМ. Причем до передачи управления обязательно необходимо запланировать следующую активность, а также модельное время ее вызова из УПМ или приостановить работу генератора.

В теории имитационного моделирования ориентированная во времени последовательность событий (активностей), отражающая поведение элемента СС, называется процессом [8, 9]. Следовательно, каждому генератору ИМ, разрабатываемой согласно БСФ МІСІС4, необходимо поставить в соответствие некоторый процесс. Поскольку переходы между активностями процесса скрыты внутри самих активностей, то при добавлении генератора в ИМ достаточно указать только начальную активность его процесса.

Аналогичным графом активностей (процессом) отображается механизм обслуживания транзакта на устройстве. При поступлении транзакта на определенное устройство УПМ вызывает начальную активность механизма обслуживания именно этого устройства. Говорят также, что УПМ инициализирует (запускает) механизм обслуживания устройства. В начальной активности можно организовать как общий, так и уникальный способ обработки различных типов транзактов. Как и для генератора, начальную активность устройства необходимо задать при его добавлении в ИМ.

В отличие от генератора каждый процесс – механизм обслуживания в конечном итоге следует завершить стандартной активностью УПМ, которая называется «Окончание обслуживания», или последней активностью. Ее назначение заключается в перемещении транзакта с исходного обслуживающего устройства на целевое, а

также в поиске транзактов для обслуживания на освободившемся исходном устройстве. Внутри последней активности возможны четыре варианта обработки транзакта:

- на исходном устройстве транзакт заканчивает передвижение по ИМ и автоматически удаляется УПМ;

- целевое устройство готово начать обслуживание транзакта, что и реализуется УПМ;

- целевое устройство не готово начать обслуживание, а исходное устройство наделено свойством сохранить транзакт до тех пор, пока целевое устройство находится в данном состоянии;

- целевое устройство не готово начать обслуживание, но исходное устройство не наделено свойством сохранения транзакта. Этот случай возникает тогда, когда разработчик ИМ не учел все аспекты взаимодействия элементов модели или ошибся при задании свойств исходного устройства. Как следствие, УПМ аварийно останавливает эксперимент с ИМ.

Таким образом, правильная обработка возникающих при завершении механизма обслуживания транзакта всевозможных вариантов обеспечивается УПМ МІСІС4. В результате разработчику ИМ достаточно сосредоточить свои усилия только на внутреннем взаимодействии транзакта и устройства, что существенно облегчает написание программы ИМ.

Как правило, в ИМ существует несколько групп однотипных по функционированию генераторов, устройств и транзактов. Каждая такая группа в МІСІС4 называется компонентом. Различие между компонентом и элементом напоминает соотношение между классом и объектом в технологии объектно-ориентированного программирования [7]. То есть компонент представляет собой некоторый шаблон

(тип), который характеризует состояние (набор значений стандартных и уникальных переменных) соответствующих ему элементов, а также связанный со статическими компонентами процесс.

Конкретные элементы любого статического компонента модели, т.е. вида устройства либо генератора, называются *версиями* устройства или генератора соответственно. Для динамических элементов, т.е. транзактов, используется термин *копия*. Таким образом, элементы одного и того же компонента модели совпадают по набору атрибутов (параметров) и процессу, а различаются по конкретным значениям, присвоенным атрибутам элементов, и следующей активности процесса. Значения как стандартных, так и уникальных атрибутов элементов сохраняются между активностями.

2. Статическая иерархическая структура имитационной модели

При декомпозиции многих СС исследователю недостаточно ее представления в виде совокупности неделимых элементов [2, 9]. Он вводит подсистемы, которые в свою очередь также могут содержать конечное количество подсистем и элементов. В MICIC4 подсистеме соответствует составное устройство, называемое *узлом*. Причем в состав узла можно вводить другие узлы, не ограничивая количество уровней вложенности.

Поскольку узел реализован как устройство, то для него также надо определить механизм обслуживания. При необходимости в одной из активностей данного процесса транзакт перемещается на определенное устройство, входящее в узел. При этом переходе механизм обслуживания на узле останавливается, пока транзакт не вернется снова на узел из любого внутреннего устройства узла. Считается, что транзакт находится одновременно и на узле

и на внутреннем устройстве этого узла. При наличии иерархической статической структуры ИМ для транзакта необходимо задавать *глубину вложенности* – максимальное количество уровней, на которое он может опуститься. По умолчанию глубина вложенности равна нулю.

Кроме попадания на узел извне транзакт может быть порожден внутри самого узла. Если требуется переместить этот транзакт за пределы узла, то первоначально он отправляется на сам узел. В этой ситуации из УПМ будет инициализирован механизм обслуживания транзакта на узле, где и реализуется его дальнейшая траектория движения по ИМ.

Компоненты, соответствующие узлам, содержат внутри себя другие компоненты – генераторы, узлы и элементарные устройства в строго фиксированном количестве. Поэтому согласно рассматриваемому способу формализации ИМ составляется *граф вложенности компонентов*. В MICIC4 на основе опыта моделирования различных СС принято ограничение, заключающееся в невозможности включения одного и того же компонента в два разных узла. Данное ограничение не является очень строгим. Оно тривиально обходится путем создания двух одинаковых компонентов, но с разными идентификаторами. Так как любой компонент может входить только в один узел или находиться на верхнем уровне иерархии, то граф статической структуры ИМ является деревом.

Из-за наличия иерархии статические элементы ИМ в MICIC4 имеют *двойную адресацию*: абсолютную и относительную. В первом случае доступ к элементу модели происходит по имени компонента и его абсолютному номеру среди всех элементов ИМ данного типа. Во втором случае областью види-

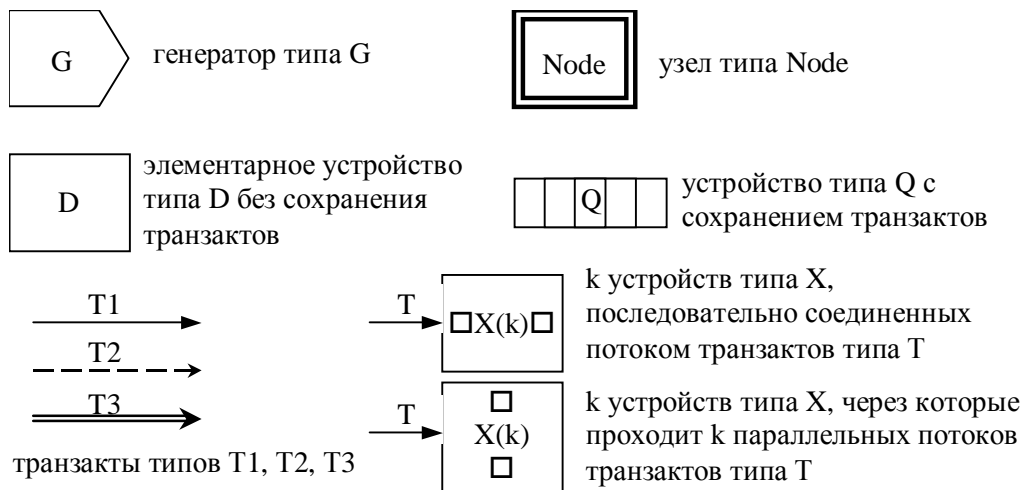


Рис. 1. Графические примитивы для создания граф-схемы ИМ

мости является только узел, в который входит статический элемент модели. Очевидно, чтобы получить доступ к элементу, являющемуся единственным в узле, достаточно указать только имя компонента. Перевод относительных адресов в абсолютные обеспечивается УПМ МІСІС4.

3. Потокосная граф-схема имитационной модели

Статическая структура ИМ позволяет отобразить свойство вложенности (подчиненности) компонентов. Но любому дереву компонентов можно поставить в соответствие произвольное количество вариантов состава статических элементов внутри узла. При этом элементы не связаны друг с другом. Следовательно, вторым аспектом декомпозиции объекта моделирования должно быть построение динамической структуры ИМ.

Она образуется путем создания связей между статическими элементами модели. Очевидно, признаком связи является возможность перемещения транзакта между элементами, т.е. отображением динамической структуры объекта моделирования служит граф-схема движения потоков транзактов по устройствам ИМ. Для ее однозначной трактовки предлагается использовать графические примитивы,

приведенные на рис. 1. Вершины граф-схемы соответствуют статическим элементам модели, а ориентированные дуги показывают возможные пути перемещения транзактов. При этом каждому типу транзакта ставится в соответствие направленная линия с уникальными параметрами оформления (толщина, штриховка, ...), как и каждому типу устройства – уникальный графический блок. Структура узла может приводиться как на одной и той же схеме, так и отдельно.

В рассматриваемой схеме формализации нет отдельного графического блока для удаления транзактов. Эта операция реализуется в рамках механизма обслуживания обычного устройства. А на граф-схеме в данной ситуации из блока, соответствующего устройству, которое удаляет транзакт, выходит меньше дуг, чем входит. В общем случае несколько одинаковых дуг, входящих в блок или выходящих из него, символизируют различные потенциальные пути движения одного и того же типа транзактов в устройство или из него. При правильном построении граф-схемы в элементарное устройство должна входить по крайней мере одна дуга, а в генератор не должна входить ни одна.

4. Особенности реализации УПМ

Главной структурой УПМ МІСІС4, обеспечивающей реализацию алгоритма организации квазипараллелизма, является массив событий. Отдельный элемент массива событий содержит следующую информацию:

- о модельном времени наступления события;
- об элементе модели (транзакте или генераторе), с которым связано событие;
- об активности, которая является обработчиком данного события.

Так как любое устройство без транзакта всегда находится в состоянии бездействия, то в массиве событий не может быть на него ссылки. С другой стороны, транзакт не может одновременно обслуживаться на двух устройствах (при передаче транзакта внутрь узла механизм обслуживания на узле прерывается). Поэтому в массиве событий достаточно хранить информацию только о генераторе или транзакте. Обслуживающее устройство может быть найдено среди атрибутов транзакта.

Массив событий упорядочен по времени наступления события. Чем раньше должно произойти событие, тем меньший номер оно имеет в массиве. Из УПМ циклически вызывается активность, соответствующая наиболее раннему событию. Очевидно, оно находится в начале массива. Этот элемент массива будем называть *начальным событием*, а связанный с ним элемент ИМ – *активным*.

В языке моделирования МІСІС4 любая активность представляет собой функцию базового универсального языка программирования и включает в себя:

- алгоритм обработки события, изменяющий состояние эле-

ментов ИМ;

- указание активности – обработчика следующего события, которое произойдет в активном элементе ИМ в зависимости от его нового состояния;

- синхронизацию, т.е. планирование времени наступления следующего события в текущем механизме обслуживания транзакта на устройстве (механизме функционирования генератора) или удаление начального события из массива в силу того, что время наступления следующего события в данном механизме определяется в других элементах ИМ;

- возврат в УПМ признака продолжения или окончания реплики (прогона) ИМ.

Операция планирования времени наступления следующего события заключается в увеличении значения модельного времени для начального события. Так как массив событий упорядочен по данному атрибуту (полю), то после сортировки, как правило, в начале массива оказывается другой элемент. Тем более такая ситуация возникает при удалении начального события из массива. Во избежание трудно локализуемых ошибок моделирования в активности после операции синхронизации должен быть только оператор возврата в УПМ.

Массивы структур также используются для работы с компонентами и всеми видами элементов модели: генераторов, устройств, транзактов. *Массив компонентов* прежде всего является отображением дерева вложенности компонентов. Помимо этого он предназначен для решения следующих задач:

- преобразования относительных адресов статических элементов в абсолютные;

- поиска статических элементов одного и того же типа

внутри соответствующих массивов устройств и генераторов;

- определения начальных активностей статических компонентов;
- задания дисциплин обслуживания и последних активностей устройств.

Массив компонентов инициализируется начальными значениями еще на этапе загрузки программы ИМ в оперативную память компьютера. Затем отдельные свойства его элементов корректируются по мере добавления генераторов и устройств в соответствующие массивы. Они должны быть сформированы до начала эксперимента с ИМ. После старта имитационного эксперимента массив компонентов доступен только для чтения, а попытка добавить еще один статический элемент приведет к ошибке выполнения.

Состояние массива транзактов, наоборот, много раз изменяется во время прогона ИМ. Транзакту при создании отводится свободный элемент массива. Далее, независимо от маршрута движения, транзакт сохраняется в этом элементе вплоть до его удаления из модели. Следовательно, для операций с транзактом достаточно знать всего лишь его индекс в массиве транзактов. В силу такой организации в нем, как правило, присутствуют пустые элементы, что следует учитывать при написании программы ИМ.

Одним из атрибутов массивов генераторов, устройств и транзактов является номер компонента, которому принадлежит элемент модели. Остальные поля массивов уникальны для каждого вида (класса) и предназначены для реализации рассматриваемого способа формализации объекта моделирования. Так как невозможно охватить все варианты взаимодействия элементов модели, то раз-

работчику программы ИМ предоставляются механизмы пополнения компонентов новыми атрибутами в рамках концепции объектно-ориентированного программирования – наследования [7]. Доступ к ним осуществляется через те же самые массивы. Далее рассматриваются свойства базовых элементов ИМ, которые, как только что отмечалось, обеспечиваются набором стандартных атрибутов и реализованными в MISIC4 алгоритмами их обработки.

5. Свойства базовых элементов MISIC4

Обслуживающее устройство. В элементе массива событий присутствует только ссылка на обслуживающийся транзакт. Устройство, на котором данный транзакт находится, является одним из атрибутом транзакта и автоматически корректируется УПМ при движении транзакта по ИМ. При этом сохраняется время рождения транзакта и обновляется время прибытия на очередное обслуживающее устройство. Если транзакт переходит на внутреннее устройство узла, то атрибуты обслуживающего устройства и времени прибытия изменяются на новые. Когда транзакт возвращается на узел, то значения данных атрибутов восстанавливаются.

Выбор маршрута движения транзактов. Непосредственное перемещение транзакта с одного (исходного) устройства на другое (целевое) реализует последняя активность механизма обслуживания исходного устройства. В любой предыдущей активности данного процесса необходимо обеспечить выбор целевого устройства для дальнейшего движения. В противном случае транзакт будет пытаться обслуживаться на том же самом исходном устройстве.

Данная концепция реализуется с помощью указателя

направлення движения по устройствам модели – стандартного атрибута, которым обладает каждый транзакт. В процессе обслуживания необходимо изменить значение указателя, чтобы определить целевое устройство. Поскольку в механизме обслуживания можно изменять значение указателя произвольное количество раз в любой активности, то справедливо говорить, что в MICIS4 реализован программируемый динамический способ движения транзакта по модели. Например, не составляет труда организовать выбор целевого устройства в зависимости от выполнения некоторого условия или по вектору вероятностей (в этом случае на граф-схеме ИМ из исходного устройства будет выходить более одной дуги с одинаковыми параметрами оформления).

Разработчику ИМ необходимо учитывать ограничения по видимости целевого устройства. Указателю можно присваивать:

- целевое устройство из того же узла, что и исходное (переход на одном уровне иерархии);
- специальное константное значение OUTSIDE. В этом случае транзакт удаляется из ИМ последней активностью;
- внутреннее устройство из узла, на котором обслуживается транзакт. Непосредственное перемещение транзакта на уровень ниже реализуется специальной функцией синхронизации. При этом активное событие с информацией о следующей активности сохраняется в транзакте и удаляется из массива событий, а УПМ запускает механизм обслуживания на внутреннем устройстве. Если последнее невозможно, то генерируется ошибка моделирования;
- узел, в который входит исходное устройство. Перемещение на уровень выше обеспечивается последней активностью исходного

устройства. Если транзакт был создан внутри целевого узла, то запускается его механизм обслуживания. В противном случае транзакт возвращается наверх и УПМ добавляет в массив событий новый элемент, восстанавливая механизм обслуживания на узле.

Количество транзактов, одновременно обслуживаемых на устройстве. В MICIS4 в любой момент модельного времени на устройстве можно одновременно обслуживать один и более транзактов либо вообще ни одного. Иногда такое свойство устройства называют многоканальностью, где одному обслуживаемому транзакту выделяется один канал устройства.

В MICIS4 используется другая концепция многоканальности. Она основана на том, что устройство имеет неотрицательный целочисленный атрибут, называемый "максимальный объем каналов V_{maxD} ", а транзакт обладает атрибутом "занимаемый объем каналов V_{occT} ". Оба указанных атрибута по умолчанию равны 1, т.е. транзакту выделяется один обслуживаемый канал устройства. Объем занятых каналов устройства V_{occD} определяется как сумма занимаемых каналов тех транзактов, которые обслуживаются на устройстве D . Очевидно, объем свободных каналов $V_{freD} = V_{maxD} - V_{occD}$. Из определения также следует, что как V_D , так и V_T могут принимать нулевые значения.

Если транзакт перемещается на условно занятое устройство, то возникает ошибка моделирования и эксперимент с ИМ завершается. Устройство D считается условно занятым для транзакта T , если выполняется следующее условие: сумма текущего объема занятых каналов устройства и занимаемого объема каналов пришедшего на обслуживание транзакта

больше максимального объема каналов данного устройства, т.е. $V_{occD} + V_{occT} > V_{maxD}$. В противном случае говорят, что устройство для транзакта свободно.

В процессе моделирования можно изменять объем каналов как у устройств, так и у транзактов. При уменьшении максимального объема каналов устройства или при увеличении занимаемого объема каналов транзактом контролируется только возможность выполнения операции. Если новое значение данного атрибута транзакта приведет к нарушению условия занятости устройства, то действие будет проигнорировано. Аналогично нельзя уменьшить максимальный объем каналов устройства до значения, меньшего, чем суммарный объем занятых каналов обслуживающимися на устройстве транзактами. При уменьшении занимаемого объема каналов у транзакта, если он ожидает обслуживания, УПМ пробует запустить механизм обслуживания устройства по указателю направления движения этого транзакта. Если увеличивается максимальный объем каналов устройства, то УПМ пытается заполнить появившиеся каналы транзактами, ожидающими обслуживания на нем.

Управление началом обслуживания по приоритету. На устройство, имеющее нулевой максимальный объем каналов, могут попасть только транзакты, занимающие нуль каналов, причем не испытывая конкуренции со стороны других транзактов. Чтобы была возможность ограничить автоматическое попадание транзактов с нулевым объемом каналов, в MISC4 введено дополнительное условие начала обслуживания транзакта на устройстве.

Устройство и транзакт наделены стандартным неотрицательным целочисленным атрибутом P_D и P_T

соответственно, который называется "приоритет". Если $P_T > P_D$, т.е. приоритет транзакта больше приоритета устройства, на которое перемещается данный транзакт, то говорят, что устройство *условно закрыто* и возникает ошибка моделирования, а эксперимент с ИМ завершается. Изменение приоритета обслуживаемого транзакта (в отличие от корректировки занимаемого им объема каналов) никоим образом не сказывается на его текущем обслуживании. Аналогично изменение приоритета обслуживаемого устройства не влияет на находящиеся на нем транзакты. Однако увеличение приоритета устройства приводит к попытке УПМ начать обслуживание тех транзактов, которые не могли попасть на данное устройство из-за своего высокого приоритета.

Очевидно, если транзакт имеет нулевой приоритет и нулевой объем каналов, то он не имеет ограничений по обслуживанию на любом устройстве. В общем случае, для начала обслуживания транзакта T на устройстве D должно выполняться следующее условие:

$$V_{occT} \leq V_{freD} \ \& \ P_T \leq P_D.$$

Варьирование объемом каналов и приоритетом позволяет эффективно реализовать взаимодействие информационных и управляющих транзактов в ИМ.

Режимы окончания обслуживания транзакта на устройстве. Устройство может быть очередью или сервером. Устройство-очередь сохраняет транзакты, перемещающиеся с него последней активностью на обслуживание на условно занятое или условно закрытое целевое устройство. Транзакты при этом переходят в ждущее состояние (из *serving* в *waitingstart*). Когда появляется возможность продолжения движения на целевое

устройство, УПМ запускает обработчик данного события, в котором разработчик ИМ может выполнить заключительные операции по обслуживанию (чаще всего это обновление статистик моделирования).

По умолчанию устройство является сервером, т.е. без сохранения транзактов. Для нормального выполнения последней активности на сервере необходимо, чтобы целевое устройство, куда перемещается транзакт, было для него свободно и открыто. Иначе возникает ошибка и эксперимент с ИМ завершается.

Частным случаем очереди является устройство, в котором транзакты сохраняются ограниченный интервал модельного времени, находясь в состоянии *waitingtime*. Если в течение данного интервала целевое устройство оказалось готово начать обслуживание, то транзакт выходит из очереди. В противном случае по истечении заданного интервала УПМ вызывает обработчик события, где необходимо либо определить новый интервал ожидания, либо перенаправить транзакт на другое устройство.

Уход транзакта с устройства. После ухода транзакта T_1 с устройства D_1 на некоторое устройство D_0 освобождается определенный объем каналов. Вследствие чего УПМ пытается найти для обслуживания транзакты, находящиеся в ждущем состоянии в очередях и чей указатель направления движения совпадает с исходным устройством D_1 , т.е. тех транзактов, которые не смогли ранее начать обрабатываться из-за недостаточного объема свободных каналов. Если находится такой транзакт T_2 , то УПМ запускает новый механизм обслуживания. При оставшемся объеме свободных кана-

лов на исходном устройстве D_1 поиск транзактов продолжается.

Следует обратить внимание, что механизмы обслуживания M_1 – транзакта T_1 на устройстве D_0 и M_2 – транзакта T_2 на устройстве D_1 начинаются в один и тот же момент модельного времени. Однако в реальном времени УПМ отдает приоритет процессу M_1 . Если требуется изменить порядок выполнения процессов, то надо в первой активности механизма обслуживания M_2 оставить состояние без изменения и запланировать следующее событие через ничтожно малый момент модельного времени. Тем самым цель будет достигнута всего лишь за счет введения «лишней» активности.

Поскольку прибывший транзакт T_2 освобождает свой объем каналов на прежнем устройстве D_2 (откуда он ушел), то УПМ пытается найти транзакты на обслуживании и для этого устройства D_2 . Такой циклический процесс продолжается, пока не будет найдено ни одного транзакта на обслуживание. Иными словами, если имеется последовательность устройств D_n, \dots, D_2, D_1 с ограниченным объемом каналов, полностью занятых транзактами, находящимися в состоянии *waitingstart* или *waitingtime*, то при уходе транзакта с устройства D_1 УПМ реализует попытку перемещения транзактов с устройства D_i на устройство D_{i-1} для всех i от 2 до n .

Очередность выбора транзактов на обслуживание. Порядок выбора транзактов на обслуживание на освободившееся или открывшееся устройство определяется дисциплиной обслуживания, представляющей собой функцию языка моделирования MICIC4. В нем реализованы стандартные дисциплины обслуживания и предоставлены механизмы для создания уникальных дисциплин разработчиком ИМ. Ос-

новая идея заключается в том, что УПМ находит пребывающие в состоянии *waitingstatrt* или *waitingtime* транзакты, указатель направления движения которых совпадает с устройством, запросившим транзакты. Если кандидатов на обслуживание оказывается больше, чем способно принять устройство, то разработчик модели с помощью дисциплины обслуживания разрешает конфликт.

При таком подходе искомые транзакты не обязательно должны находиться в одной и той же очереди. Однако невозможно обеспечить обратное, т.е. переход транзактов на обслуживание из общей очереди на любое устройство из группы устройств $DD = \{D_1, D_2, \dots, D_n\}$, которое освободилось первым. Чтобы снять данное ограничение, каждое устройство имеет стандартный атрибут, называемый *направляемым устройством*. Этот атрибут для группы DD должен указывать на любое устройство из DD (но одно и то же!). Пусть для определенности это будет устройство D_1 . Если транзакт из общей очереди не может продолжить обслуживание ни на одном из устройств группы, то его указателю направления движения необходимо присвоить значение направляемого устройства D_1 . Как только любое из устройств группы DD освободится, то УПМ будет искать кандидатов на обслуживание также среди транзактов, ждущих освобождения направляемого устройства D_1 .

Таким образом, в MISIC4 отсутствуют ограничения на порядок выбора транзактов на обслуживание из одной или более очередей.

Внешние события для транзакта. Практически в каждой ИМ приходится нарушать тривиальное обслуживание транзакта на устройстве, заключающееся в синхронизации с УПМ по временным за-

держкам. Например, может понадобиться прервать обслуживание при или до наступления определенного события; организовать параллельное движение нескольких транзактов с независимым или подчиненным управлением; сначала расщепить транзакт на несколько потоков, а затем собрать их вместе и т.п. Эти операции реализуются через прием транзактом следующих сигналов: «Задержать», «Отпустить», «Исключить», «Уничтожить».

Сигнал «Задержать» переводит транзакт в состояние *delayed*, перемещая соответствующее транзакту событие из массива событий в область стандартных атрибутов транзакта. Тем самым выполнение механизма обслуживания откладывается на неопределенный интервал модельного времени до получения транзактом сигнала «Отпустить». Данный сигнал возвращает событие в массив, и УПМ передает управление восстановленной активности в заданный момент модельного времени. Эти сигналы подаются именно на транзакт, а не на устройство, чтобы исключить неоднозначность, связанную с многоканальностью устройства.

В случае получения сигнала «Уничтожить» обслуживание транзакта прерывается и он поглощается УПМ. Также удаляется элемент массива событий, соответствующий транзакту, если он находится в состоянии *servng*. В результате уничтожения транзакта освобождаются соответствующие каналы как обслуживавшего устройства, так и внешних узлов, через которые он попал на данное устройство. Как следствие, реализуется попытка принять на обслуживание на эти устройства новые транзакты.

Сигнал «Исключить» переводит транзакт в состояние *"excluded"*, также прерывая об-

служивание транзакта на устройстве с высвобождением занятых каналов обслуживавшего устройства и внешних узлов, но не удаляя транзакт из ИМ. И только этим он отличается от сигнала «Уничтожить». Время перехода транзакта в данное состояние заносится в его атрибут "время начала обслуживания". Далее в определенный момент исключенный транзакт можно снова отправить на обслуживание, т.е. в состояние "serving", таким же точно образом, как впервые созданный транзакт.

Если транзакт задерживает, исключает или уничтожает самого себя, то из массива событий удаляется начальный элемент. Как отмечалось выше, во избежание ошибки моделирования после любого из данных действий в активности необходимо сразу же возвратиться в УПМ.

Внешние события для генератора. Генератор может принимать сигналы: «Пассивизировать», «Активизировать». Первый сигнал временно запрещает выполнение очередной активности процесса, убирая ее из массива событий. При этом в свойствах генератора сохраняется следующая выполняемая активность. Если генератор переводит в пассивное состояние самого себя, то данное действие в активности должно быть последним. Второй сигнал, соответст-

венно, возвращает отложенную активность генератора в массив событий и передает на нее управление в заданный момент модельного времени. Первоначально в ИМ генератор добавляется в пассивном состоянии. Фактически, эта пара сигналов является полным аналогом сигналов для транзакта «Задержать»/«Отпустить».

Организация стохастического потока внешних событий посредством генератора. Как отмечалось выше, генераторы предназначены для организации потоков внешних событий в ИМ. Причем в подавляющем большинстве случаев данные потоки являются стохастическими [2, 9]. Поэтому одним из стандартных свойств генератора является номер датчика случайных чисел, который будет давать реализации случайной величины согласно требуемому закону распределения. Если для генератора требуется более одного потока, то их необходимо реализовать через уникальные свойства генератора.

Уникальные атрибуты элементов ИМ. Если для отражения функционирования элементов модели стандартных атрибутов недостаточно, то устройства, транзакты и генераторы дополняются множествами уникальных атрибутов.

Обработка значений уникальных атрибутов реализуется внутри активностей с помощью всего арсенала средств базового языка программирования ИМ. Важная особенность уникальных атрибутов заключается в сохранении состояния элементов между различными активностями одного и того же процесса.

6. Графическое отображение механизмов обслуживания

Механизм обслуживания транзакта на устройстве схематически задается *графом переходов между активностями*. При этом рекомендуется придерживаться обозначений, представленных на рис. 2. Вершины соответствуют активностям, а маркированные направленные дуги – переходам между активностями и внешним управляющим сигналам.

Механизм обслуживания в любом случае содержит по крайней мере две активности: начальную и конечную. У устройства-сервера они соединены одинарной сплошной дугой, а у очереди – пунктирной.

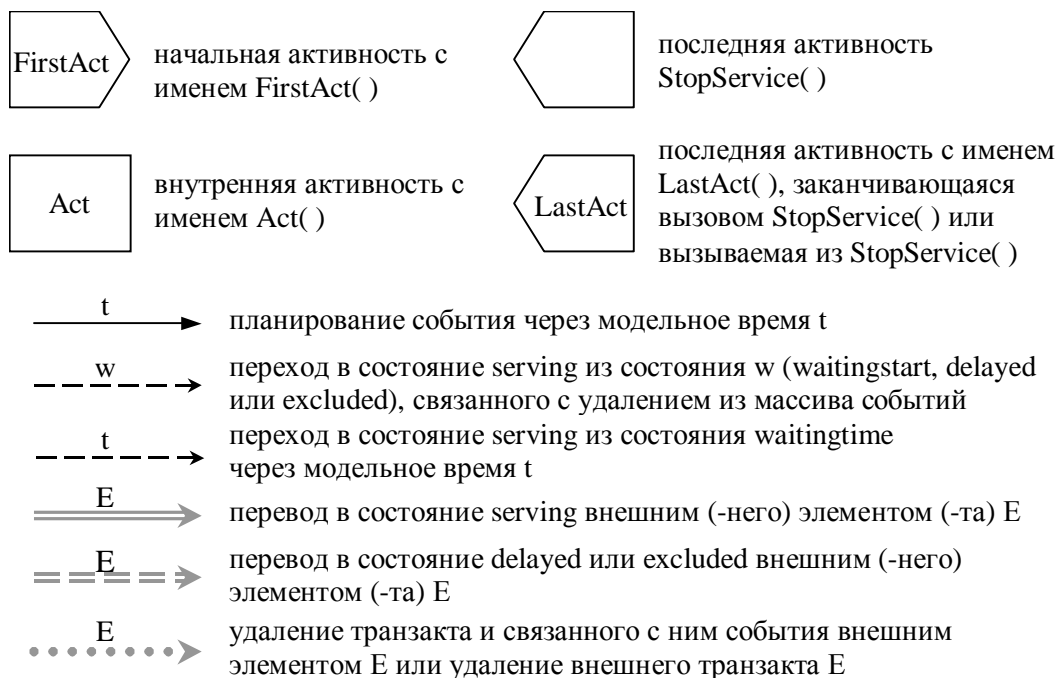


Рис. 2. Условные обозначения при отображении механизма обслуживания транзакта на устройстве

Двойные серые дуги являются висячими и предназначены для указания управляющих сигналов в направлении других элементов ИМ.

Любой исходящей дуге должна соответствовать хотя бы одна входящая дуга в другом механизме обслуживания транзакта (функционирования генератора) и наоборот. Данные дуги помечаются идентификаторами внешних элементов. Аналогично отображается удаление транзакта из ИМ другим ее элементом.

Для механизма функционирования генератора используются те же самые графические примитивы. Активному состоянию генератора соответствует состояние транзакта *serviing*, а пассивному — *delayed*. Очевидно, что в механизмах функционирования генераторов отсутствует последняя активность.

Выводы

Описанная концепция структуризации подсистем и элементов СС и представление динамики их взаимодействия с помощью вышеприведенных базовых конструкций носит название формализации объекта моделирования согласно БСФ МІСІС4. Методика формализации включает следующую последовательность этапов:

- определение иерархической статической структуры объекта моделирования;
- выделение динамических потоков транзактов;
- построение сетевой граф-схемы ИМ на каждом уровне иерархии;
- определение уникальных атрибутов всех компонентов модели;
- графическое отображение механизмов функционирования генераторов и обслуживания транзактов на устройствах;

- алгоритмизация активно-стей механизмов.

Овладение данной методикой необходимо для написания программы ИМ на языке моделирования MICIS4, который предоставляет взаимно-однозначный программный интерфейс, позволяющий автоматизировать дискретно-событийное моделирование СС на высоком (достаточно) уровне детализации.

1. Задачи и модели ИСО. Ч.3. Технология имитации на ЭВМ и принятие решений: Уч. пособие/ И.В. Максимей, В.Д. Левчук и др. – Гомель: БелГУТ, 1999. – 150 с.
2. Технология системного моделирования/ Е.Ф. Аврамчук, А.А. Вавилов, С.В. Емельянов и др. Под общ. ред. С.В. Емельянова, В.В. Калашникова и др. — М.: Машиностроение, 1988. — 520 с.
3. Рыжиков Ю.И. Имитационное моделирование. Теория и технологии. — М.: Альтекс-А, 2004. — 384 с.
4. Шрайбер Т.Дж. Моделирование на GPSS. — М.: Машиностроение, 1980. — 592 с.
5. Томашевский В.Н., Жданова Е.Г. Имитационное моделирование в среде GPSS. — М.: Бестселлер, 2003. — 416 с.
6. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II. — М.: Мир, 1987. — 646 с.
7. Страуструп Б. Язык программирования С++: 3-е изд. — М.: БИНОМ, 1999. — 991 с.
8. Шеннон Р. Имитационное моделирование систем: искусство и наука. — М.: Мир, 1978. — 418 с.
9. Лоу А., Кельтон В. Имитационное моделирование. Классика СС: 3-е изд. — СПб.: Питер, 2004. — 847 с.

Получено 29.09.04

Об авторе

Левчук Виктор Дмитриевич,
доцент, канд. техн. наук

Место работы автора:

Гомельский государственный университет им. Ф. Скорины, кафедра математических проблем управления
246699, Гомель, ул. Победы, 27а/31
Тел. 55 7828 (дом.), 56 4237 (раб.)