

**ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ НОВОЙ СРЕДЫ
ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ****Д. А. Герасимов, С. А. Олейникова, О. Я. Кравец (Воронеж)**

Рассматривается задача построения новой среды имитационного моделирования. В [1–3] приводились общие принципы и особенности такой среды. После этого возникла необходимость учесть выделенные особенности при программной реализации новой системы. Для того чтобы максимально упростить процесс построения моделей, исследователь должен оперировать простейшими объектами, функциональность которых будет предоставлена ему средой [4]. Таким образом, для построения эффективного средства имитационного моделирования необходимо описать элементарный объект, с которым предстоит взаимодействовать пользователю. Рассмотрим, какой функциональностью должна обладать данная сущность. Для этого выделим задачи, которые должен решить исследователь в процессе построения модели, и задачи, возлагаемые на среду имитационного моделирования. Представим их в виде двух списков.

1. Задачи, решаемые исследователем:
 - а) задание структуры модели;
 - б) формализация условий прекращения моделирования.
2. Задачи, решаемые системой:
 - а) мониторинг событий, возникающих в системе;
 - б) выполнение обработчиков событий;
 - в) обеспечение взаимодействия элементов модели.

Рассмотрим более подробно каждый пункт представленного списка. Задание структуры модели сводится к созданию пользователем всех элементов модели, а также установкой связей между ними. К этому этапу также можно отнести указание значений параметров элементов модели, соответствующих исследуемой области. Структура модели в готовом виде должна представлять собой совокупность связанных элементов с заданными параметрами и, таким образом, будет являться входными данными модели. Автоматизировать данный процесс не представляется возможным, так как заранее не известны проблемы, которые необходимо решить исследователю. Поэтому этот этап построения модели выполняется пользователем самостоятельно. Однако в целях облегчения работы исследователя следует ввести несколько следующих принципов: создание элемента сводится к выбору его типа и места расположения; создание связи сводится к указанию двух связываемых элементов; вводимые параметры проверяются на соответствие их назначению, и если обнаруживаются ошибки, об этом незамедлительно извещается исследователь. Таким образом, следуя перечисленным принципам, можно максимально упростить рассматриваемый этап.

Формализация условий прекращения моделирования также возлагается на пользователя и сводится к заданию логических условий, при выполнении которых система останавливает процесс моделирования и предоставляет статистические данные эксперимента. В подавляющем большинстве имитационных моделей используются два типа таких условий: достижение определённого модельного времени и достижение количества вошедших в модель или же покинувших её транзактов. Однако иногда требования исследуемой области таковы, что прекращение моделирования должно происходить по условию, не принадлежащему ни одному из двух приведённых типов. Например, если необходимо остановить процесс имитации, когда очередь перед какой-либо сущностью достигнет определённой величины. Исходя из этого, представляется нецелесообразным ограничить круг условий двумя самыми распространёнными типами. Следует дать исследователю в руки механизм, способный удовлетворить требованиям любой предмет-

ной области. В соответствии с этим необходимо разработать систему так, чтобы пользователь мог задать условия, связанные с любой сущностью модели, не применяя при этом каких-либо специальных средств.

Далее рассмотрим задачи, решаемые самой системой, т.е. задачи, подлежащие автоматизации. Мониторинг событий, возникающих в системе, необходим для того, чтобы выделить в процессе моделирования ключевые моменты, в которые может возникнуть необходимость выполнить какие-либо действия. Собственно, эти ключевые моменты и есть события. Система должна вести список этих событий, который, помимо информации о типе события, должен включать необходимое количество параметров, а также данные об элементе модели, для которого событие предназначено. Само событие представляет собой совокупность условий наступления и списка параметров. Формирование условий наступления, также как и условий окончания моделирования, должно быть доступно для любых сущностей модели. Это позволит выделить практически любые ключевые моменты в процессе имитации.

Задача исполнения обработчиков событий тесно связана с задачей мониторинга. Сами по себе события не представляются полезными. Каждый элемент должен иметь возможность произвести какие-либо действия при наступлении некоторых, заранее заданных, условий. Таким образом, имеет место понятие реакции элементов. Обработчик события представляет собой инструкции на встроенном языке высокого уровня, которые направлены на то, чтобы поведение элемента выглядело адекватным предметной области. В общем случае исследователю не придётся задавать этот код. Под задачей исполнения обработчиков событий понимается следующий список действий: нахождение соответствующего обработчика при наступлении какого-либо события, передача ему параметров этого события, исполнение кода на встроенном языке высокого уровня. Отметим, что в результате исполнения обработчика в список событий могут быть добавлены новые элементы, которые, в свою очередь, также являясь событиями, приведут к исполнению соответствующих обработчиков [3].

Самой сложной в реализации задачей является обеспечение взаимодействия элементов. Прежде всего под этим понимается то, что связанные элементы должны быть способны передавать друг другу транзакты. Для уменьшения ошибок имитации представляется целесообразным осуществлять передачу путём посылки некоторого запроса. Принимающий элемент решает сам, имеет ли он возможность в данный момент забрать транзакт, и если да, то осуществляет это действие [3]. Это, например, позволит избежать случая, когда количество транзактов в элементе превышает количество каналов обслуживания. Кроме передачи транзактов элементы должны быть способны автоматически их создавать и удалять. Под созданием в данном случае понимается действие, которое приводит к появлению нового транзакта, а также к заполнению всех его параметров, если они есть. При удалении осуществляется корректное уничтожение соответствующей программной единицы. При пересылке транзакта в некоторых видах моделей необходимо указывать, какой именно из них должен быть передан. Чаще всего порядок передачи осуществляется по методу FIFO, однако для организации других дисциплин необходимо иметь возможность адресовать каждый транзакт в элемент. В этих целях необходимо организовать список транзактов внутри элемента. В таком случае адресация будет происходить по номеру транзакта в списке. Сами элементы модели также могут иметь параметры, например количество каналов обслуживания или максимальную длину очереди. Кроме таких статических параметров, которые не изменяются в течение всего процесса моделирования, у элементов могут быть и динамические свойства. В соответствии с этим у исследователя должен быть механизм, позволяющий менять значения параметров элемента, однако это может привести к ошибкам имитации и получению неверных статистических данных. Чтобы максимально сократить количе-

ство таких ошибок, предлагается ввести принцип инкапсуляции, который будет действовать между элементами модели [1]. Это означает, что никакой элемент извне напрямую не может изменить свойство другого элемента. Для того чтобы этого достигнуть, необходимо послать запрос на изменение, а принимающий элемент, свойство которого нужно изменить, в соответствии со своим внутренним состоянием и принципом адекватности предметной области сам изменит (или же нет) значение нужного параметра. Это, например, позволит избежать случая, когда элемент, имеющий определённое количество каналов на одном участке процесса моделирования, меняет это количество на другом этапе. Естественно полагать, что иногда возникает необходимость в переменном количестве каналов, когда, например, некоторые из них выходят из строя (уменьшение), или же когда необходимо ввести в работу новый резервный канал (увеличение), однако такие возможности как раз и включает в себя принцип адекватности.

Таким образом, разделив задачи, решаемые системой, и задачи, решаемые пользователем, а также рассмотрев их более подробно, можно в общем виде описать элементарный объект, который позволит максимально упростить процесс построения модели. Для организации мониторинга событий необходимо, чтобы у элементарного объекта присутствовал механизм, позволяющий сохранять список возможных, т.е. всех зарегистрированных в системе, событий и список текущих событий, в котором ведётся учёт параметров конкретных событий и вызовов конкретных обработчиков. В соответствии с иерархической структурой модели [1] список таких событий для текущего уровня абстракции может храниться у родительского объекта. Все элементы текущего уровня также будут обладать такими списками, но, не имея внутри себя подуровней, списки таких элементов будут пустыми.

Чтобы организовать механизм реакции элементов на события, необходимо разработать инструмент, благодаря которому исследователь может определить обработчик конкретного события. Для достижения этой цели предлагается использовать методы, которые, по сути, являются именованными частями кода на встроенном языке высокого уровня. Информацию о соответствии конкретного метода и конкретного события необходимо добавить в список возможных событий, благодаря чему метод приобретает свойства обработчика. Для выполнения каких-либо действий, не связанных напрямую с событиями, например, для расчёта динамического приоритета и пересортировки списка транзактов, предлагается использовать методы, которые не регистрируются в системе как обработчики событий и существуют отдельно от них. Вызов такого метода может быть осуществлён в любом месте кода, в том числе и в обработчиках каких-либо событий.

Ранее было показано [2], что транзакты и элементы модели целесообразно не делить на две разные сущности, а представить одной, которая обладает признаком статичности для таких элементов модели как, например, устройства или очереди, или же признаком динамичности для элементов, исполняющих роль транзактов. В соответствии с этим у элементарного объекта должен быть список транзактов, который представляет динамичные элементы, а также поименованные поля, которые представляют статичные объекты. Т.е. если объект относится к классу динамичных, то он может фигурировать в соответствующих списках всех элементов текущего уровня модели. Передача какого-либо члена списка определённого элемента модели в список другого элемента и будет являться передачей транзакта. Добавление и удаление элементов таких списков будут соответствовать добавлению и удалению транзактов. Как указывалось ранее, статичные элементы будут представлены поименованными полями. Целесообразнее всего разметить такие поля внутри родительского объекта. Это позволит, с одной стороны, открыть доступ родителю ко всем элементам следующего уровня, а с другой – ограничит доступ элементов одного уровня друг к другу. Для того чтобы связан-

ные элементы могли осуществлять взаимодействие друг с другом, т.е. посылать запросы на передачу транзактов, запросы на изменение параметров элемента и т.д., необходимо, чтобы в каждом элементе в виде тех же именованных полей присутствовали копии ссылок на связанный объект, который размещён в родительском элементе.

В ходе работы были решены следующие задачи:

1. Определены основные этапы, которые должен выполнить исследователь при работе с новой средой имитационного моделирования и функции, выполняемые самой системой;

2. Детализированы задачи, выполнение которых необходимо для обеспечения взаимодействия элементов в данной среде;

3. Разобран процесс передачи транзакта от одного элемента другому;

4. Описан общий подход реакции элементов на события.

Следующим шагом является практическая реализация разрабатываемой среды на основании предложенных ранее принципов и приведенных в данной работе особенностей.

Литература

1. Герасимов Д. А. Об одном способе построения имитационных моделей // Компьютерные технологии в технике экономике: сборник докладов международной научной конференции. Ч. 1. Воронеж: МИКТ, 2007. С. 11–15.
2. Герасимов Д. А., Кравец О. Я. Статичные и динамичные элементы в разрабатываемой среде имитационного моделирования // Современные проблемы информатизации в проектировании и информационных системах: Сб. трудов. Вып.14. Воронеж: Научная книга, 2009. С. 467–468.
3. Герасимов Д. А., Олейникова С. А. Среда построения имитационных моделей // Территория науки. 3(4). Воронеж: Научная книга, 2007. С. 333–337.
4. Советов Б. Я., Яковлев С. А. Моделирование систем 3-е издание. М.: Высшая школа, 2001. 343 с.