

**РАЗВИТИЕ ТЕОРИИ АГРЕГАТИВНЫХ СИСТЕМ – КАНОНИЧЕСКАЯ ФОРМА
ИМИТАЦИОННОЙ МОДЕЛИ****В. Г. Хлопяк (Москва)****Введение**

В настоящее время процесс разработки имитационных моделей исследуемых систем во многом остается творчеством, результат зависит от квалификации разработчиков и часто носит случайный характер. В первую очередь это связано с отсутствием технологий формализации (описания моделирующего алгоритма) сложных систем.

Наиболее общим подходом к формализованному описанию сложных систем является подход, базирующийся на понятиях агрегата и агрегативных систем [1, 2, 4]. Под агрегатом понимается динамическая система с дискретным вмешательством случая. Агрегативными считаются системы, каждый элемент которых представляет собой агрегат, а передача информации между ними происходит мгновенно и без искажений.

Однако представление объектов реального мира в виде столь общей математической схемы (А-схемы) требует немалого искусства, большого опыта и доступно лишь узкому кругу специалистов. По мере увеличения сложности моделируемой системы значительно возрастает трудоемкость, заметно снижается оперативность и рентабельность моделирования и, как следствие, уменьшается сфера практического применения [1, 3].

Кроме того, существенным недостатком является зависимость разработанной А-схемы от количества агрегатов в системе. Как и для большинства имитационных моделей, изменение числа агрегатов приводит к необходимости разработки новой А-схемы.

В данной работе рассматривается метод построения имитационных моделей сложных систем в виде А-схем, который не имеет указанных недостатков.

Снижение сложности разработки модели достигается путем применения более простой, по сравнению с классической, модели «Х-агрегата», а зависимость А-схемы от количества агрегатов устраняется изменением механизма взаимодействия агрегатов.

Модель «Х-агрегата»

Определение «Х-агрегата» строится исходя из общих представлений о характере функционирования объектов реального мира.

1. Х-агрегат существует и функционирует во времени и пространстве. Процесс функционирования Х-агрегата рассматривается как последовательность действий, выполняемых агрегатом под воздействием внешних сигналов (сообщений).

2. Х-агрегат обладает некоторым набором свойств $A = \{a_1, a_2, \dots, a_n, z\}$ и имеет конечное множество («особых») состояний $Z = \{z_1, z_2, \dots, z_n\}$. Набор свойств является постоянным на всем интервале моделирования. Каждое свойство a_i на интервале моделирования может принимать значения из некоторого непустого множества допустимых значений $dom a_i$. Свойство z предназначено для фиксации («особого») состояния объекта $dom z = Z$. Состояние объекта в момент времени $t \in (0, T)$ обозначается как $z(t)$, $z(t) \in Z$. В пространстве объект представляется совокупностью значений его свойств (в памяти ЭВМ объект занимает некоторое место, выделенное для записи значений его свойств).

3. Х-агрегат имеет один вход для приема входных сообщений. Сообщения поступают в дискретные моменты времени. В один момент времени агрегат может принимать и обрабатывать только одно сообщение. До окончания обработки текущего со-

общения поступление других сообщений считается недопустимым. Выполнение данного ограничения должно обеспечиваться механизмом взаимодействия агрегатов.

Множество классов входных сообщений U конечно и делится на две группы: управляющие и информационные $U = \{E, L\}$, где $E = \{e_0, e_1, e_2, \dots, e_m\}$ – множество классов входных управляющих сообщений; $L = \{l_1, l_2, \dots, l_r\}$ – множество классов входных информационных сообщений.

Входное сообщение, поступающее в момент $t_i \in (0, T)$, обозначается как $u(t_i)$. Все входные сообщения имеют одну и ту же структуру и представляются упорядоченной парой $u = \langle u_n, u_d \rangle$, где u_n – имя сообщения, u_d – содержание (данные) сообщения. Имя сообщения определяет класс, к которому принадлежит данное сообщение, – $u_n \in U$.

4. X-агрегат имеет один выход для передачи выходных сообщений. Если в один и тот же момент времени агрегат должен передать несколько сообщений, то передача осуществляется последовательно в заранее определенном порядке.

Множество классов выходных сообщений Y конечно и аналогично множеству входных сообщений делится на два типа: управляющие и информационные $Y = \{E^*, L^*\}$, где $E^* = \{e_1^*, e_2^*, \dots, e_p^*\}$ – множество классов выходных управляющих сообщений; $L^* = \{l_1^*, l_2^*, \dots, l_p^*\}$ – множество классов выходных информационных сообщений.

Структура выходного сообщения зависит от его типа.

Выходные управляющие сообщения представляются упорядоченной четверкой:

$$e^* = \langle ID_i, y_n, y_d, \Delta t \rangle,$$

где ID_i – идентификатор агрегата, являющегося приемником (адресатом) сообщения;

y_n – имя выходного сообщения, $y_n \in E^*$;

y_d – содержание (данные) сообщения;

Δt – задержка сообщения.

Задержка сообщения определяется как разность между моментом его отправки агрегатом-источником и моментом его получения агрегатом-приемником. Задержка сообщения обеспечивается моделью взаимодействия объектов. Определяя задержку сообщения как случайную величину с заданным законом распределения, можно моделировать случайный характер взаимодействия объектов.

Выходные информационные сообщения представляются упорядоченной четверкой вида

$$l^* = \langle ID_i, y_n, y_d, y_r \rangle,$$

где y_r – содержание ответа на сообщение (возвращаемое значение).

Посредством управляющих сообщений реализуется асинхронное взаимодействие агрегатов, посредством информационных – синхронное.

5. X-агрегат рассматривается как дискретная детерминированная динамическая система без последействия. При получении входного сообщения агрегат выполняет определенное действие $d^* \in D$ из заданного конечного множества

$$D = \{d_1, d_2, \dots, d_n, d_0, d_x, q_1, q_2, \dots, q_r\}.$$

Действие выполняется мгновенно и зависит от класса входного сообщения.

Формально реакция агрегата на входное сообщение задается оператором F . Если в момент t_i поступило входное сообщение $u(t_i) = \langle u_n, u_d \rangle$, то действие агрегата d^* определяется по следующей формуле:

$$d^*(t_i + 0) = \begin{cases} q_k, & u_n = l_k; \\ d_j, & u_n = e_k \ \& \ F(z(t_i), u_n) = z_j; \\ d_0, & u_n = e_k \ \& \ F(z(t_i), u_n) = 0; \\ d_x, & u_n = e_k \ \& \ F(z(t_i), u_n) = x. \end{cases}$$

Таким образом, при получении информационного сообщения агрегат выполняет действие, соответствующее классу сообщения вне зависимости от состояния, в котором он находится.

Реакция агрегата на управляющее сообщение зависит от его состояния и может быть трех видов: собственно действие – d_j , игнорирование сообщения – d_0 и “ошибка” – d_x . Действие d_x предназначено для контроля функционирования модели, содержит операцию выдачи оператору сообщения об ошибке и операцию “Аварийный останов”.

Действие представляет собой набор последовательно исполняемых элементарных операций (примитивов). Последовательность исполнения операций определяется логикой действия. Операции делятся на группы: вычисления, проверка условий, организация циклов, передача сообщений, окончание действия, аварийный останов и выдача сообщения лицу, проводящему эксперимент. Отличие типов действий состоит в том, что X-агрегат может изменять состояние и отправлять управляющие сообщения только при выполнении действий типа $d_j \in D$.

При выполнении действия d_j первой операцией является установка нового состояния агрегата

$$z(t_i + 0) = z_j.$$

Так как действие выполняется мгновенно, то переход агрегата из одного состояния в другое происходит скачком, а собственно действие выполняется при нахождении агрегата в новом состоянии.

Каждому состоянию соответствует уникальный набор операций, т. е. каждому состоянию агрегата соответствует одно уникальное действие.

Оператор F , применительно к реакции на управляющее сообщение, может быть задан аналогично способу задания работы конечных автоматов – в виде таблиц переходов и состояний [4].

Механизм взаимодействия агрегатов

Основное требование, предъявляемое к механизму взаимодействия, – обеспечение инвариантности модели по отношению к числу агрегатов системы. Например,

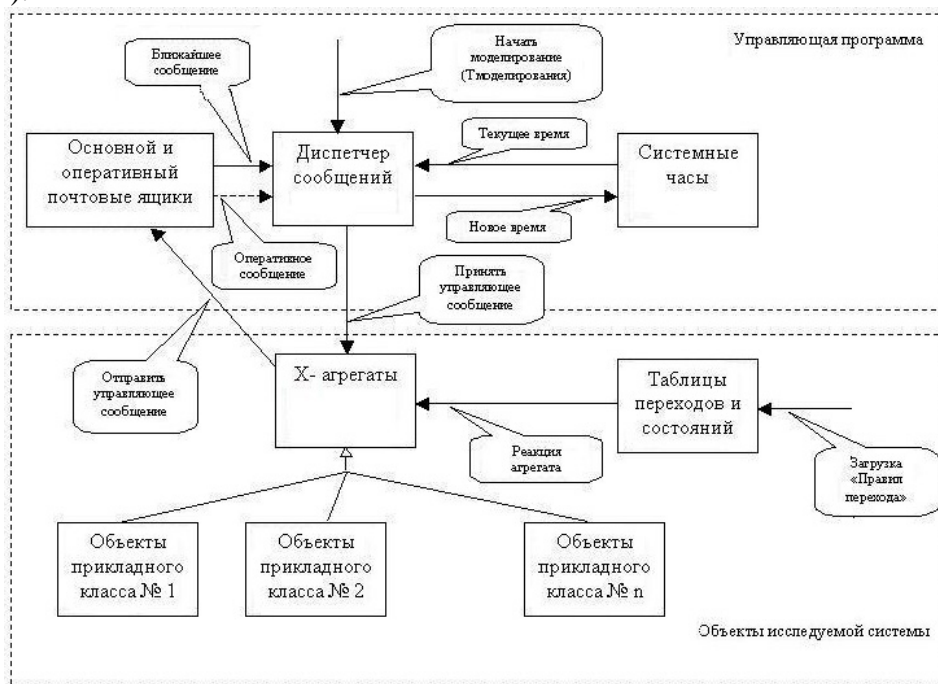
применительно к моделям систем массового обслуживания (СМО), механизм взаимодействия должен обеспечивать независимость моделей от количества каналов обслуживания, т. е. одноканальная и многоканальная СМО представляются в виде одной и той же модели – модели с изменяемым числом каналов.

Введем понятие канонической формы имитационной модели: **имитационная модель системы представлена в канонической форме, если она не зависит от количества элементов (агрегатов) системы.**

Данная проблема непосредственно связана с другой сложной проблемой имитационного моделирования – проверкой адекватности моделей.

Решение рассматриваемой проблемы осуществляется путем включения в имитационную модель специального объекта – диспетчера сообщений.

Работу диспетчера сообщений поясняет типовая А-схема имитационной модели (см. рисунок).



Типовая А-схема моделируемой системы

Диспетчер управляет передачей сообщений между агрегатами и непосредственно связан с системными часами. “Течение” времени осуществляется по принципу особых состояний Δz .

Диспетчер “оснащается” двумя почтовыми ящиками, предназначенными для приема и хранения сообщений, отправляемых агрегатами.

В соответствии с принятыми предположениями о структуре сообщений, каждое сообщение помимо информационной части имеет адресную. В адресной части указывается идентификатор приемника сообщения, имя сообщения и время получения (“вручения”) сообщения адресатом.

Работа диспетчера, связанная с обработкой сообщений, сводится к следующему.

1. Сначала диспетчер отыскивает и удаляет из основного почтового ящика “ближайшее” сообщение (ближайшим считается сообщение, в котором время “вручения” имеет минимальное значение).

2. После этого, используя время “вручения”, указанное в ближайшем сообщении, «переводит» системные часы.

3. Далее, используя информацию, находящуюся в адресной части сообщения, доводит сообщение до адресата и ожидает ответа об окончании его обработки адресатом. В имитационной программе доведение сообщения реализуется вызовом (передачей управления) соответствующей функции агрегата. Завершение функции означает окончание обработки сообщения адресатом.

При обработке любого из сообщений агрегат, перейдя в новое состояние, может посылать (генерировать) выходные сообщения. Некоторые сообщения являются отложенными и поступают в основной почтовый ящик. Другие сообщения должны быть обработаны в текущий момент времени и их наличие обусловлено требованиями согласованного перехода в новое состояние других объектов. Например, в модели СМО переход агрегата, относящегося к классу «Заявки», в состояние «Обслуживание» связан с переходом в тот же момент времени некоторого агрегата из класса «Каналы обслуживания» в состояние «Работа». Такие сообщения имеют время задержки, равное нулю, и поступают в «оперативный» почтовый ящик.

4. После получения ответа об окончании обработки текущего сообщения агрегатом диспетчер проверяет наличие сообщений в оперативном почтовом ящике. В случае наличия оперативного сообщения диспетчер обрабатывает его по той же схеме, что и основное сообщение. В случае, если оперативный почтовый ящик пуст, диспетчер приступает к поиску следующего «ближайшего» сообщения в основном почтовом ящике.

5. Работа диспетчера сообщений продолжается до тех пор, пока системные часы не покажут время окончания моделирования или основной почтовый ящик не окажется пустым.

В предложенном механизме взаимодействия синхронизация работы различных объектов системы достигается путем разделения процесса взаимодействия на два этапа – отправку сообщения источником и получение сообщения приемником. При реализации моделирующего алгоритма на ЭВМ отправка сообщения реализуется вызовом функции создания объекта – сообщения (вызовом конструктора класса сообщений). Получение сообщения реализуется вызовом функции обработки сообщения X-агрегатом.

В случае реализации моделирующего алгоритма на ЭВМ последовательного действия никаких дополнительных элементов синхронизации в алгоритм вводить нет необходимости. В любой момент времени активным является только один объект и ему доступны все ресурсы ЭВМ. Имитация одновременного функционирования нескольких объектов достигается за счет применения оперативного почтового ящика и управления системным временем.

При реализации алгоритма на ЭВМ параллельного действия классы объектов могут функционировать на различных вычислителях. В алгоритмах, предусматривающих использование нескольких процессоров, потребуется проверка готовности ресурса, выделенного для функционирования объектов.

Вывод

Предлагаемые модели X-агрегата и диспетчера сообщений позволяют разработать имитационную модель достаточно сложной системы, представив ее в канонической форме. Реализация модели в виде имитационной программы возможна на любом объектно-ориентированном языке на основе применения механизма наследования и перегрузки функций [5].

Литература

1. **Бусленко Н. П.** Моделирование сложных систем. М.: –Наука, 1968.
2. **Бусленко Н. П., Калашников В. В., Коваленко И. Н.** Лекции по теории сложных систем. –М.: Советское радио, 1973.
3. **Бусленко В. Н.** Автоматизация моделирования сложных систем. –М.: Наука, 1977.
4. **Советов Б. Я., Яковлев С. А.** Моделирование систем. М.: Высшая школа, 1985.
5. **Буч Г.** Объектно-ориентированное проектирование с примерами применения. –М.: Конкорд, 1992.