

Формирование имитационного процесса на основе алгоритмического описания функционирования информационной системы

77-30569/291975

11, ноябрь 2011

Черненький В. М.

УДК 004.436.4

МГТУ им. Н.Э. Баумана
chernen@bmstu.ru

Поскольку изложение материала статьи опирается на определение алгоритмической модели процесса, кратко изложим содержание этого понятия.

Процесс Z есть четверка [1]:

$$Z = \langle S, T, F, \alpha \rangle$$

где:

S - пространство состояний системы, определенное ранее;

T - множество моментов времени изменения состояний системы;

F – график процесса, определяемый как отображение $T \rightarrow S$, причем это отображение должно быть функциональным (однозначным);

α - отношение линейного порядка на T .

Пусть имеем объект O_i в системе Q . Тогда генерация процесса Z_{O_i} может быть выполнена путем задания оператора H^{O_i} [1]:

где: $t_i \in T_{O_i}$;

A - множество аргументов: $A \subseteq Q$;

ω - случайное число.

Остальные определения опираются на материал [2].

В ходе развития процесса совокупность аргументов во множестве A^{O_l} может изменяться. Таким образом, состав элементов множества A^{O_l} в общем случае зависит от времени. Обозначим эту зависимость как $A_{t_i}^{O_l}$. Рассмотрим два объекта O_l и O_m в системе Q . Пусть $O_l \cap O_m = \emptyset$, а процессы в них заданы следующими операторами:

$$s_{t_i}^{O_l} = H^{O_l}(A_{t_i}^{O_l}, t_i, \omega); \quad s_{t_i}^{O_m} = H^{O_m}(A_{t_i}^{O_m}, t_i, \omega).$$

Если $O_m \cap A_{t_i}^{O_l} = \emptyset$ и $O_l \cap A_{t_i}^{O_m} = \emptyset$, то такие процессы и объекты называются *не сцепленными* в момент времени t_i .

Если $O_l \cap A_{t_i}^{O_m} \neq \emptyset$, то объект O_m *сцеплен* с объектом O_l в момент времени t_i . То же относится и к их процессам. Это означает, что для определения состояния объекта O_m в момент времени t_i , необходимо знание состояния объекта O_l в это же время. Обозначим отношение сцепления как $O_l \rightarrow O_m$. Не следует смешивать отношение сцепления и зависимости. Так, если $O_m \rightarrow O_l$ и $O_l \rightarrow O_k$, то вовсе не обязательно, чтобы $O_m \rightarrow O_k$. Таким образом, отношение сцепления не является транзитивным.

Если к отношению сцепления добавить полное транзитивное замыкание, то получим *отношение зависимости*. Т.е. если O_l зависит от O_m , а O_k зависит от O_l , то O_k зависит и от O_m . Таким образом, *отношение сцепления можно определить как отношение непосредственной зависимости*.

Предлагается задавать оператор H в виде некоторой алгоритмической структуры.

Рассмотрим *дискретный во времени* процесс Z . Пространство состояний S может быть как непрерывным, так и дискретным. Поставим в соответствие каждой i -ой точке процесса (момент времени изменения состояния t_i) некоторый оператор h_i^c . Оператор h_i^c вычисляет значение состояния $s_i \in S$ в момент времени t_i :

$$s_i = h_i^c(A_i, t_i, \omega).$$

Оператор h_i^c описывает вычисление только одной i -й точки процесса Z . В силу этого условия будем в дальнейшем называть этот оператор *элементарным*.

Таким образом, если график процесса содержит n точек, то мы должны задать линейную последовательность элементарных операторов:

$$h_1^c, h_2^c, \dots, h_i^c, \dots, h_n^c.$$

Введем новый элемент модели - *инициатор*. Он обладает свойствами динамичности и инициативности. Динамичность - инициатор имеет возможность перемещаться от оператора к оператору; будем называть попадание инициатора на оператор *сцеплением инициатора с элементарным оператором*. Инициативность - в момент сцепления инициатора с оператором происходит *выполнение (инициирование) элементарного оператора*, что соответствует вычислению нового состояния процесса. При этом полагаем, что выполнение элементарного оператора происходит *мгновенно*. Таким образом, описание процесса может быть выполнено путем задания линейной последовательности операторов $\langle h_i^c \rangle_{i=1}^n$ и перемещения по этой последовательности инициатора I , сцепляющегося с элементарными операторами h_i^c в заданные моменты времени t_i изменения состояния процесса. Алгоритмическая модель описания процесса предполагает, что *моменты сцепления инициатора с элементарными операторами определяют сами элементарные операторы*. С этой целью вводится *оператор условия сцепления инициатора* h_i^y , который определяет условие, при выполнении которого инициатор сцепляется со следующим оператором h_{i+1}^c . Возможны следующие варианты задания такого условия:

- а) указание момента времени сцепления инициатора с оператором h_{i+1}^c ;

б) определение логического условия, при выполнении которого инициатор сцепляется с оператором h_{i+1}^c .

Таким образом, определим элементарный оператор h_i , как двойку:

$$h_i = \langle h_i^c, h_i^y \rangle.$$

При сцеплении инициатора с элементарным оператором h_i происходит мгновенное выполнение его обеих составных частей: выполнение h_i^c позволяет вычислить новое состояние s_i процесса Z , а выполнение оператора h_i^y дает возможность определить момент времени, либо логическое условие сцепления инициатора со следующим элементарным оператором h_{i+1} . Линейную последовательность элементарных операторов назовем *треком* TR :

$$TR = \langle \{h_i\}_{i=1}^n, \beta \rangle.$$

Таким образом, можно АМП определить как двойку:

$$\text{АМП} = \langle TR, I \rangle.$$

Построение имитационной модели информационной системы выполняется на основе алгоритмического описания процессов функционирования, таким образом, модель описания параллельных процессов позволяет выполнить исследование в области методов построения имитационных моделей. Имитационный процесс рассматривается, как один последовательный вычислительный процесс, на который отображается система параллельных взаимосвязанных процессов.

Пусть заданы два элементарных оператора h_l и h_k одного процесса Z , причем $h_l \rightarrow h_k$, (h_k сцеплен с h_l).

Утверждение 1. Если $h_l \rightarrow h_k$, то

а) $t_k \geq t_l$

б) первым должен вычисляться оператор h_l .

Рассмотрим влияние отношения сцепления на последовательность выполнения операторов при моделировании. Пусть заданы два процесса Z_1 и Z_2 , условно изображенные на рис. 1.

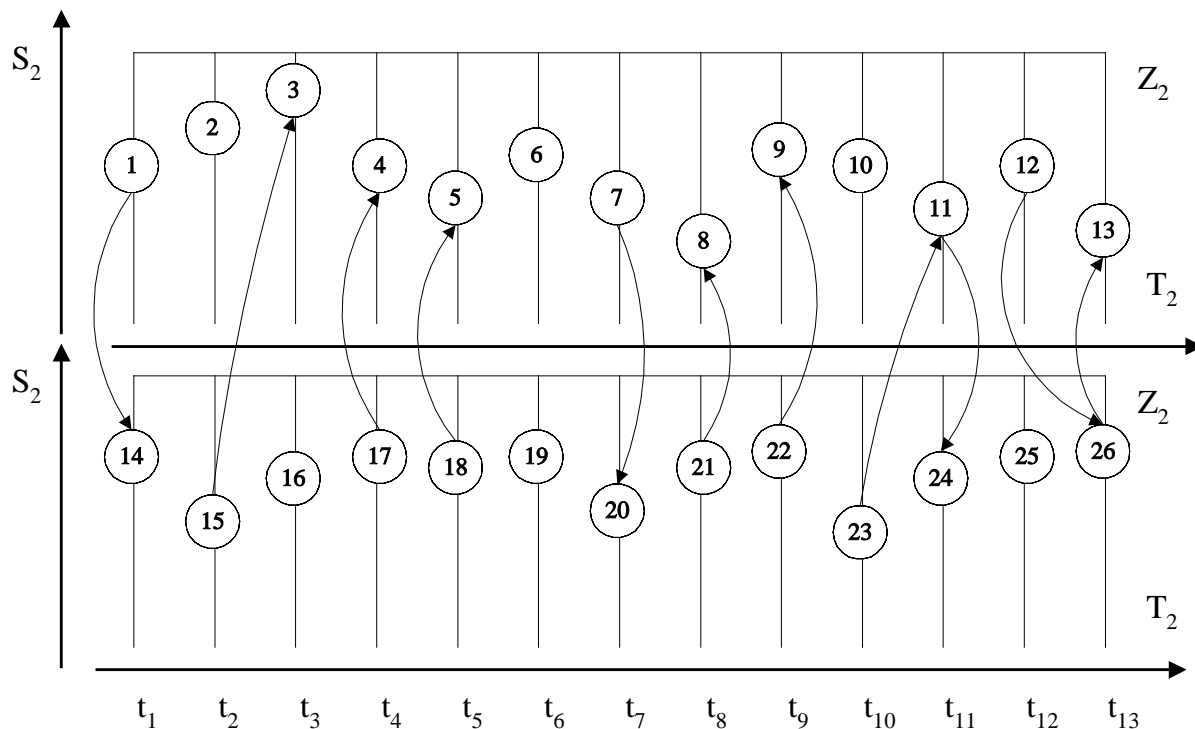


Рисунок 1. Пример сцепления процессов

Дискретные состояния процесса Z_1 пронумерованы от 1 до 13, а процесса Z_2 - от 14 до 26. Рассмотрим четыре типовых случая:

- а) моделируется один процесс Z_1 , причем его операторы не сцеплены между собой.
- б) моделируется процесс Z_1 без ограничения на сцепленность элементарных операторов;
- в) моделируются два процесса Z_1 и Z_2 , при этом эти процессы не сцеплены между собой;
- г) моделируются два сцепленных процесса Z_1 и Z_1

На рис. 1 стрелками указано отношение сцепления.

А. Моделирование процесса Z_1 при несцепленных операторах $h_i (\forall i)$.

Поскольку сцепление $h_i \rightarrow h_{i+1}$ отсутствует для всех i , то последовательность вычислений элементарных операторов не имеет значения и операторы $h_i (\forall i)$ могут вычисляться в любом порядке.

Б. Моделирование процесса Z_1 в общем случае наличия сцепленности.

Естественно предположить, что последовательные состояния одного процесса сцеплены между собой.

Предположим, что $h_i \rightarrow h_{i+1}$, для всех $i = \overline{1, n}$. Из утверждения 1 следует, что в таком случае $t_{i+1} > t_i$ для всех $i = \overline{1, n}$. Таким образом, последовательность сцепленных операторов строго следует порядку α на T . Из этого же утверждения следует, что последовательность вычислений операторов должна определяться этим же порядком.

В. Моделирование несцепленных между собой процессов Z_1 и Z_2 .

Предполагаем, что процессы Z_1 и Z_2 в отдельности представлены общим случаем. Если h_i^1 - операторы процесса Z_1 , а h_j^2 - операторы процесса Z_2 , то по предположению отсутствует сцепление между h_i^1 и h_j^2 для всех i и j . Так же, как и в случае **А**, здесь последовательность вычислений элементарных операторов из разных процессов не имеет значения. Однако, поскольку все h_i^1 сцеплены между собой, и все h_j^2 также сцеплены между собой, важно, чтобы в этой последовательности выполнялся порядок α_1 для процесса Z_1 и α_2 для процесса Z_2 . В частности, возможен вариант вычисления сначала всех операторов процесса Z_1 , а затем всех операторов процесса Z_2 . Очевидно, что порядок α_1 и α_2 при этом сохраняется.

Г. Моделирование сцепленных между собой процессов Z_1 и Z_2 . Пример сцепления показан на рис. 1.

В этом случае должен быть обязательно выдержан порядок α_1 для Z_1 и α_2 для Z_2 , поскольку в общем случае внутри каждого процесса существует сцепленность элементарных операторов. Кроме того, необходимо выполнить условия Утверждения 1 для любой пары сцепленных операторов из разных

процессов. Так, для приведенного на рисунке 1 примера в соответствии с условиями Утверждения 1, из операторов h_1 и h_{14} первым должен вычисляться оператор h_1 , из h_5 и h_{18} первым - h_{18} и т.д.

Получим возможный порядок вычислений: $h_1, h_{14}, (h_2, h_{15}), (h_3, h_{16}), h_{17}, h_4$ и т.д.

В скобках указаны пары операторов из разных процессов, для которых можно поменять порядок вычислений.

На основании проведенного анализа можно сделать следующие выводы:

1. Для каждого процесса в ходе вычисления операторов необходимо строго придерживаться порядка α на T .

2. Выполнение п.1. обеспечивает реализацию сцепленности операторов $h_i \rightarrow h_j$, если $t_i < t_j$, в рамках реализации одного процесса.

3. При выполнении п.1 указание сцепленности операторов с разным значением t_i в рамках одного процесса не добавляет новых ограничений на порядок расчета и может быть опущено.

Действительно, если вычислены элементарные операторы h_{10} и h_{23} в момент времени t_{10} , то указание $h_{23} \rightarrow h_{11}$ не меняет порядок вычислений, поскольку к моменту времени t_{11} будет рассчитан оператор h_{23} . При этом условие сцепленности выполняется автоматически.

4. Для определения порядка расчета операторов, принадлежащих разным процессам, важно знание отношения сцепления для операторов разных процессов с одинаковым значением времени t_i .

Таким образом, можно сформулировать следующий алгоритм определения порядка вычисления элементарных операторов для совокупности параллельных процессов.

Пусть заданы треки процессов Z^i ($i = \overline{1, n}$): $\langle \{h_j^i\}, \beta^i \rangle$ для всех i . Пусть текущее значение времени равно t и все элементарные операторы h^i , у которых $t^i \leq t$, вычислены. Для всех h_j^i , имеющих $t^i = t$ и обладающих условным временным оператором $(h_j^i)^{\mathcal{J}}$, определим для каждого очередной

момент времени t_{j+1}^i сцепления инициатора по своему треку, задаваемый этим временным оператором. Получим множество $\{t_{j+1}^i\}$. Назовем его *активным временным множеством*.

Это множество содержит по одному значению от каждого процесса, остановленного на элементарном операторе, содержащем временное условие продвижения инициатора. Определим очередное значение времени по формуле:

$$t_0 = \min t_{j+1}^i, (\forall i) \quad (1)$$

Последовательное применение формулы строит упорядоченное множество $\langle T^t, \alpha^t \rangle$, где α^t -линейный порядок на T^t . Докажем ряд его свойств.

Для процесса Z^i имеем множество T^i и линейный порядок α^i ($i = \overline{1, n}$). Рассмотрим множество $T = \bigcup T^i (\forall i)$ и порядок α^T на T , полученный транзитивным замыканием порядков α^i .

Утверждение 2. Множества T^t и T равны.

В самом деле, любой элемент $t \in T^t$ был определен из множества $\{t_{j+1}^i\}$, элементы которого принадлежат T^i ($i = \overline{1, n}$). Поскольку $T = \bigcup T^i (\forall i)$, то они же принадлежат и T . Следовательно, каждый элемент множества T^t принадлежит и множеству T . Аналогично доказывается, что если элемент принадлежит T , то он же принадлежит и T^t . Таким образом, утверждение 2 доказано.

Утверждение 3. На упорядоченном множестве $\langle T^t, \alpha^t \rangle$ сохраняются все порядки α^i ($i = \overline{1, n}$).

Рассмотрим множество T^i с порядком α^i . Возьмем два любых его элемента t_k^i и t_1^i . Пусть в соответствии с α^i $t_k^i < t_1^i$. Тогда, в соответствии с (2.14) t_k^i окажется минимальным элементом в $\{t_{j+1}^i\}$ раньше, чем t_1^i , и следовательно, войдет во множество T^t раньше, чем t_1^i . Поскольку

упорядочение в T^i определяется порядком поступления элементов в T^i , то, следовательно, и в T^i $t_k^i < t_1^i$. Аналогично можно провести доказательство и для остальных процессов. Таким образом, утверждение 3 доказано.

На основании Утверждений 2 и 3 можно сделать заключение о том, что формула 1 позволяет построить новое множество T , включающее все элементы множеств T^i ($\forall i$), и провести упорядочение его элементов таким образом, что сохраняются все отношения порядка в каждом из них. Поскольку каждому $t \in T^i$ ($\forall i$) ставится в соответствие свой элементарный оператор в треке, то определение порядка на T дает возможность однозначно определить порядок на всем множестве элементарных операторов заданных процессов.

Рассмотрим последовательность выполнения элементарных операторов в системе. Пусть система содержит 9 объектов, в каждом из которых развивается процесс. На рис. 2 приведен пример организации треков на некотором временном интервале. Пусть в некоторый момент времени система находилась в состоянии, показанном на рис. 2 слева. Назовем его начальным состоянием.

Начальное состояние

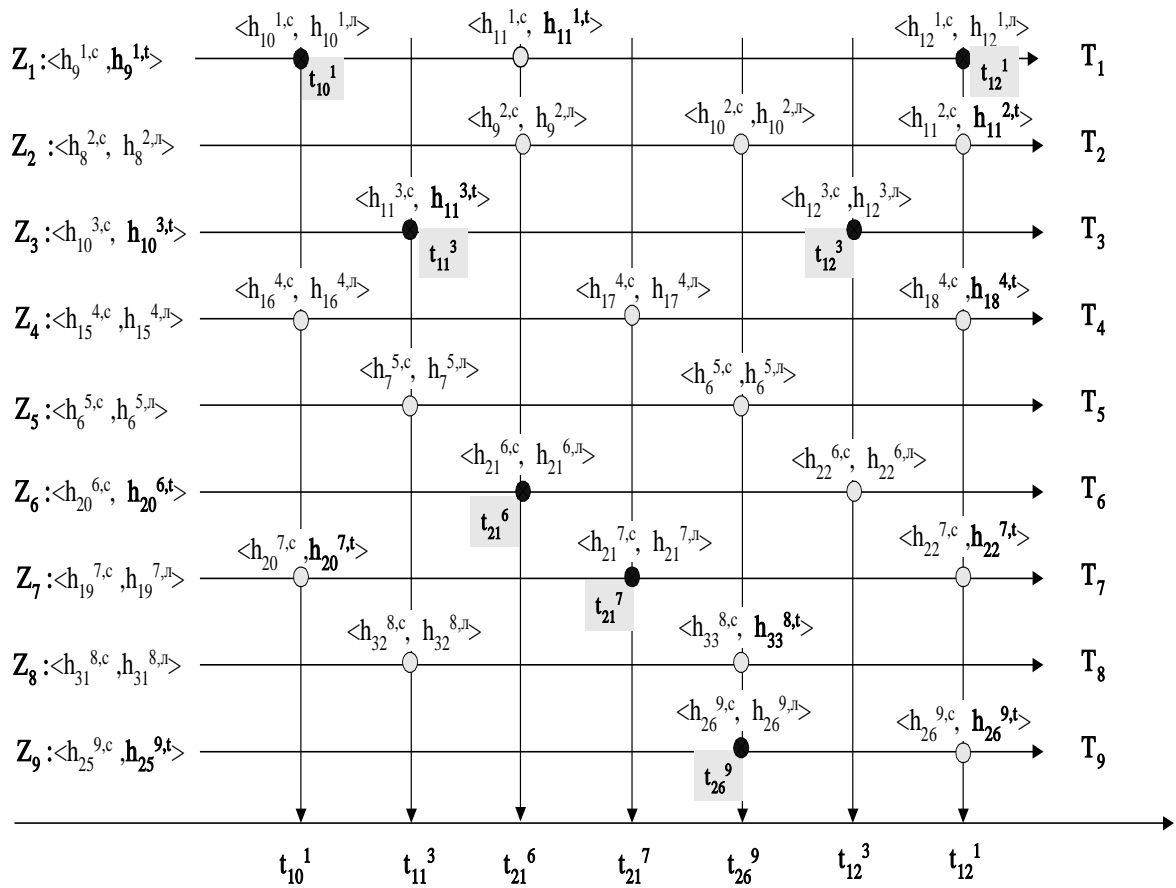


Рисунок 2. Пример реализации процессов через события

Для каждого процесса Z_i указан текущий элементарный оператор в следующих обозначениях:

$$\langle h_n^{i,c}, h_n^{i,t} \rangle,$$

где i - номер процесса (он же- номер строки);

n - порядковый номер элементарного оператора в своем треке;

c - символ “состояние”;

$u=l$ - символ логического условия продвижения инициатора;

$u=t$ - символ временного условия продвижения инициатора.

Предположим, что к этому моменту времени все элементарные операторы вычислены. Тогда активное временное множество на текущий момент равно $\{t_{10}^1, t_{11}^3, t_{21}^6, t_{26}^9\}$

Определим в соответствии с (1) очередной момент времени, как:

$$t_0 = \min \{t_{10}^1, t_{11}^3, t_{21}^6, t_{26}^9\}$$

Для примера рис. 2 $t_0 = t_{10}^1$.

Таким образом, из начального состояния система переходит в новое состояние в момент времени t_{10}^1 , соответствующее временному условию $h_{10}^{1,t}$.

В этот момент времени инициатор I_1 процесса Z_1 перемещается в элементарный оператор $(h_{10}^{1,c}, h_{10}^{1,l})$, вычисляя новое состояние объекта O_1 , а значит, и системы в целом. При этом оказывается выполненным условие $h_{15}^{4,l}$ и $h_{19}^{7,l}$. В треке процесса Z_4 выполняется оператор состояния $h_{16}^{4,c}$ и устанавливается логическое условие $h_{16}^{4,l}$, а в треке процесса Z_7 – $h_{20}^{7,c}$ и временное условие $h_{20}^{7,t}$. На рис. 2 соответствующие события отмечены светлыми кружками. Событие по временному условию показано на рис. 6 в виде темного кружка. Так как больше условий не выполняется, то необходимо перейти к новым элементарным операторам в соответствии с треками. В рассматриваемый момент времени следующее активное временное множество имеет вид $\{t_{11}^3, t_{21}^6, t_{21}^7, t_{26}^9\}$. Согласно (1) необходимо перейти к новому моменту времени $t_0 = \min\{t_{11}^3, t_{21}^6, t_{21}^7, t_{26}^9\}$. В нашем случае $t_0 = t_{11}^3$. Первым выполняется оператор $\langle h_{11}^{3,c}, h_{11}^{3,t} \rangle$, который изменяет состояние объекта O_3 и системы в целом, а также формирует новый заказ времени передвижения своего инициатора I_3 в момент времени t_{12}^3 . При этом выполняются логические условия для процессов Z_5 и Z_8 . Инициаторы I_5 и I_8 вызывают выполнение операторов $\langle h_{7}^{5,c}, h_{7}^{5,l} \rangle$ и $\langle h_{32}^{8,c}, h_{32}^{8,l} \rangle$ соответственно. Формируется новое активное временное множество: $\{t_{12}^3, t_{21}^6, t_{21}^7, t_{26}^9\}$, наименьшим в котором является t_{21}^6 . С оператора $(h_{21}^{6,c}, h_{21}^{6,l})$ и начинается следующий цикл вычислений, и т.д.

На этом примере хорошо иллюстрируется алгоритм продвижения модельного времени и порядок выполнения элементарных операторов в каждом процессе. Сделаем некоторые обобщения.

Выполнение каждого элементарного оператора назовем *событием* в системе. *Событие активное, если оно следует в треке за элементарным оператором, содержащем h^t* . На рис. 2 условия, содержащие h^t , выделены жирным шрифтом. К активным событиям относятся выполнение $\langle h_{10}^{1,c}, h_{10}^{1,l} \rangle$, $\langle h_{11}^{3,c}, h_{11}^{3,t} \rangle$, $\langle h_{21}^{6,c}, h_{21}^{6,l} \rangle$ и т.д.

Событие пассивное, если оно следует в треке за элементарным оператором, содержащем h^l . На рис. 2 к пассивным событиям относятся выполнение $\langle h_{11}^{1,c}, h_{11}^{1,t} \rangle$, $\langle h_9^{2,c}, h_9^{2,l} \rangle$, $\langle h_{17}^{4,c}, h_{17}^{4,l} \rangle$ и т.д.

Множество событий, происходящих в один и тот же момент модельного времени, назовем *классом одновременных событий* (КОС). На рис. 2 класс одновременных событий в момент времени t_{10}^l включает события для 1, 4 и 7-го процессов; в момент времени t_{11}^3 КОС составляют события для 3, 5 и 8-го процессов и т.д.

Рассмотрение КОС примера на рис. 2 показывает, что в каждом КОС содержится активное событие. Введем следующие допущения:

Допущение 1. В каждом КОС содержится одно и только одно активное событие.

- Тогда:
- а) все остальные события в КОС, если они есть, являются пассивными;
 - б) количество КОС равно мощности объединенного множества времен T .

Допущение 2. Первым событием в любом КОС является активное событие.

Если это так, то оставшиеся пассивные события для определения последовательности своего выполнения требуют знания отношения сцепления. Представим отношение сцепления в КОС в виде направленного графа. На рис. 3 приведен пример некоторого КОС с заданным на нем отношением сцепления.

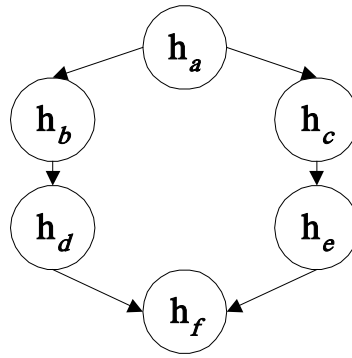


Рисунок 3. Пример КОС

Здесь: h_a - активное событие; h_b, h_c, h_d, h_e, h_f - пассивные события.

Из графа следует, что после h_a необходимо выполнить h_b либо h_c ; затем h_d , либо h_e ; и наконец h_f .

Задание отношения сцепления для каждого КОС является самостоятельной задачей. Однако можно предложить таким образом определять условие выполнения каждого пассивного события, чтобы оно содержало условие выполнения предыдущего события, т.е. содержало описание отношения сцепления с предыдущим событием.

Так, условие выполнения события h_d должно содержать выражение, проверяющее выполнение события h_b . Если это удастся сделать, то алгоритм формирования КОС выглядит следующим образом:

1. Выполняется активное событие.
2. Проверяются условия всех возможных в системе пассивных событий.
3. Если выполняется условие пассивного события, то оно вычисляется.

Алгоритм продолжается с п.2.

Важно найти признак, по которому можно определить завершенность КОС.

Теорема А

КОС завершен, если все условия, заданные операторами h^l во всех треках, равны 0.

Поскольку КОС содержит одно активное событие и оно выполняется первым, то все остальные события пассивные. Пассивное событие по определению выполняется, если определяющее его условие равно 1. Однако, по предположению, все условия, заданные h^i , равны 0. Таким образом, выполнение пассивного события невозможно, а единственное активное событие уже выполнено. Поскольку не выполняется ни один элементарный оператор, состояние системы и параметры условных операторов не могут быть изменены. Состояние системы зафиксировано и не будет изменено вплоть до нового КОС. Что и требовалось доказать.

Теорема Б

Первым событием в КОС является активное событие.

Доказывая теорему А, мы показали, что когда завершен КОС, то все условия в условных операторах системы равны 0, и состояние системы не может быть изменено ни одним пассивным событием. Значит, следующий КОС может начаться только с активного события. Что и требовалось доказать.

Таким образом, наше допущение 2 о начальной функции активного события в КОС доказано строго в теореме Б.

Литература

1. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. - М.: Сов.Радио, 1973. - 438 с.
2. Черненький В.М. Процессно-ориентированная концепция системного моделирования АСУ: диссертация ... доктора технических наук: 05.13.06, М., 2000 - 299 с.ил.

The forming of simulation process, based on algorithmic description of information system's functioning

77-30569/291975

11, November 2011

Chernen'kii V.M.

Bauman Moscow State Technical University
chern@bmsu.ru

This article deals with the basic principles of simulation process' organizing, based on the description of discrete system's functioning as a set of connected series-parallel processes, represented as algorithmic models. Fundamentals of algorithmic description are summarized. Influence of the connection of processes and operators on the execution order of elementary operators in simulation process is analyzed. The concept of active temporary set was introduced, several of its properties were proved. Also author introduced the concept of action as the fact that an elementary operator has worked out. The actions were classified by their activity in time. The concept of class of simultaneous events at the forming of simulation process appeared to be very useful and important. Properties of class and rules of its creating were proved.

Publications with keywords: [simulation process](#), [discrete systems](#), [series-parallel processes](#), [parallel process display on simulation one](#)

Publications with words: [simulation process](#), [discrete systems](#), [series-parallel processes](#), [parallel process display on simulation one](#)

Reference

1. Buslenko N.P., Kalashnikov V.V., Kovalenko I.N., Lectures on the theory of complex systems, Moscow, Sov.Radio, 1973, 438 p.
2. Chernen'kii V.M., The process - oriented concept of system modeling ACS (Dr.Sci.Tech. dissertation), Moscow, 2000, 299 p.