

**Интеграция алгоритмических описаний
функционирования системы****77-30569/292913**

12, декабрь 2011

Черненко В. М.

УДК 004.636.4

МГТУ им. Н.Э. Баумана

chernen@bmstu.ru

Поскольку предлагаемая структура алгоритмов генерации опирается на алгоритмическую модель описания процессов (АМП), то кратко изложим ее основные положения [2]. Всю совокупность параметров системы, определяющих процесс функционирования или участвующих в нем, назовем *параметрическим множеством системы* $Q = \{q_i\}_{i=1}^n$, где q_i – некоторый параметр. Каждый параметр q_i принимает множество значений, обозначаемое в дальнейшем как $\sigma(q_i)$.

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 . Назовем этот оператор элементарным, поскольку он вычисляет одно значение состояния $s_i \in S$ в один заданный момент времени t_i .

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*. Описание процесса может быть выполнено путем задания трека и перемещения по нему 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, окончательно 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. Линейную последовательность элементарных операторов назовем треком TR:

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

Таким образом, можно АМП определить как двойку:

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

В соответствии с [1] будем задавать элементарный оператор, как:

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

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

ω - случайное число.

Операторно-параметрическая схема.

Элементарный оператор h оперирует с параметрами и изменяет состояние объекта O . По отношению к оператору параметры могут быть входными, выходными и рабочими (рис. 1).

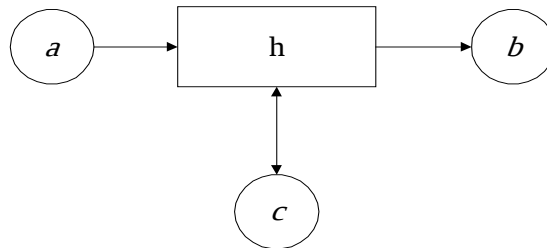


Рисунок 1. Отношение параметра к оператору

где:

a - входной параметр;

b - выходной параметр;

c - рабочий параметр.

Входной параметр означает его принадлежность к множеству A , выходной - к формированию состояния s , что говорит о его принадлежности к множеству O , рабочий - к тому и другому множеству одновременно.

Если на трек операторов указать используемые этими операторами параметры и их взаимосвязи, то получим операторно-параметрическую схему. Пример такой схемы приведен на рис. 2. Двойными линиями показан путь инициатора, а одинарными - отношение параметров к операторам. Такие схемы дают наглядную картину взаимодействия параметров в ходе реализации процесса.

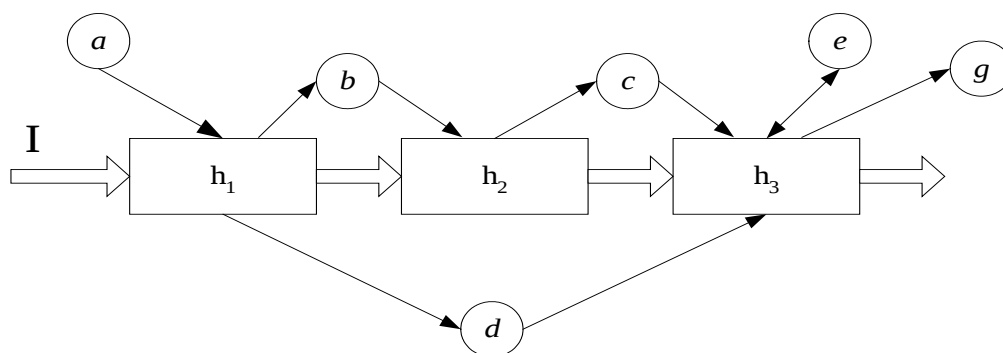


Рисунок 2. Операторно-параметрическая схема

Из схемы видно, что трек состоит из трех элементарных операторов и, следовательно, описывает три точки процесса. Из схемы следует, что для оператора h_1 множества $A_1=\{a\}$, $O_1=\{b,d\}$; для оператора h_2 множества $A_2=\{b\}$, $O_2=\{c\}$; для оператора h_3 множества $A_3=\{d,c,e\}$, $O_3=\{e,g\}$. Оператор h_2 сцеплен с оператором h_1 , поскольку параметр b является выходным для h_1 (принадлежит A_1) и входным для h_2 (принадлежит O_2), h_3 сцеплен с h_1 и h_2 через параметры d и c соответственно.

Как видно из приведенного примера, операторно-параметрическая схема дает достаточно полное представление о способе описания процесса с использованием АМП.

Описание подобных процессов.

Рассмотрим случай задания двух достаточно близких по описанию процессов Z_1 (трек А) и Z_2 (трек В) (рис. 3). И в том и в другом треке используются операторы h_1 и h_2 , но они взаимодействуют с разными параметрами как входными, так и выходными. Было бы желательно найти способ объединения описаний таких процессов. Для решения поставленной задачи дополним определение инициатора, добавив к его фундаментальным свойствам возможность включать в себя параметры. Таким образом, инициатор наряду с фундаментальными свойствами приобретает некоторое “тело” в виде совокупности параметров. Параметры в этой совокупности должны быть упорядочены. Назовем эту совокупность *локальной средой* процесса. Будем в дальнейшем считать, что “тело” инициатора представляет

собой *ссылку на локальную среду*. Таким образом, движение инициатора по треку есть движение ссылки на локальную среду по треку, а сама локальная среда может быть неподвижна. Поскольку процесс и его инициатор связаны между собой взаимно-однозначно, то мы в дальнейшем будем рассматривать выражения “локальная среда процесса” и “локальная среда инициатора” как синонимы.

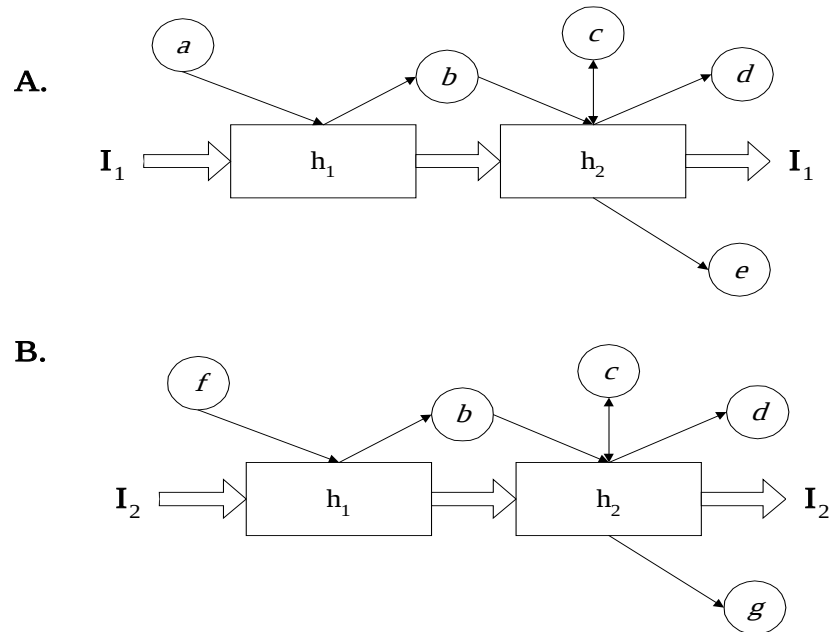


Рисунок 3. Пример подобных описаний процессов

Тогда можно предложить следующую схему свертки описаний двух процессов (рис. 3) в одно общее описание (рис. 4):

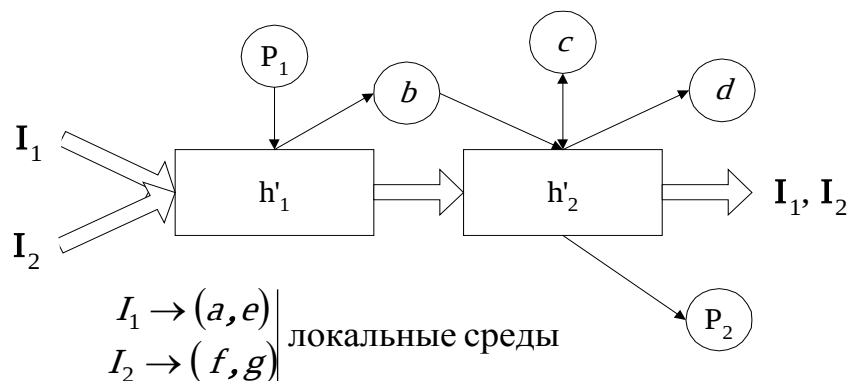


Рисунок 4. Объединенное описание процессов Z_1 и Z_2

Как видно из рисунков, с инициатором I_1 связана локальная среда (a, e), а с инициатором I_2 - локальная среда (f, g). Оператор h_1 модифицирован в оператор h'_1 , который связан с параметром b и первым параметром локальной среды инициатора. Оператор h'_2 связан с параметрами b, c, d и вторым параметром локальной среды инициатора. Операторы h'_1 и h'_2 будем называть *объединенными*. Отличие объединенных элементарных операторов от простых элементарных операторов состоит в возможности взаимодействовать не только с явно заданными параметрами, но и с параметрами локальных сред инициаторов. Инициаторы I_1 и I_2 присутствуют в этой схеме одновременно.

Очевидно, эти рассуждения могут быть распространены на случай n параллельно протекающих процессов. Процессы, сгенерированные треком или структурой, использующими объединенные элементарные операторы и локальные среды называются *подобными*.

Таким образом, удалось снизить размерность описания множества процессов, введя отношение подобия процессов. Для описания совокупности подобных процессов достаточно иметь одно объединенное описание трека или структуры и множество одинаково структурированных локальных сред, привязанных к инициаторам.

Обобщенные операторы, вложенность, блоки

На треке можно задать некоторое плотное разбиение элементарных операторов на подмножества. Это разбиение обычно выполняется с целью получения функционально однородных подмножеств операторов. Совокупность операторов, входящих в одно подмножество, назовем обобщенным оператором.

Пример разбиения приведен на рис. 5. Здесь подмножество операторов h_1, h_2, h_3 объединено в один обобщенный оператор H_1 , а h_4, h_5 - в обобщенный оператор H_2 . В результате получаем трек обобщенных операторов.

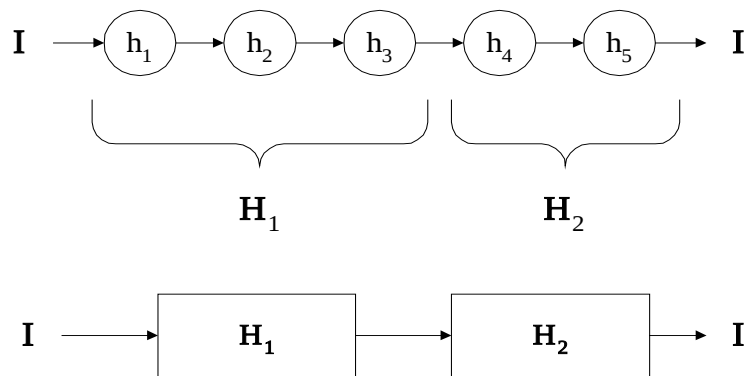


Рисунок 5. К понятию обобщенного оператора

Плотное разбиение может быть выполнено и на треке обобщенных операторов. Рекурсивность определения позволяет вводить понятия обобщенных операторов различного уровня: первого, второго и т.д.

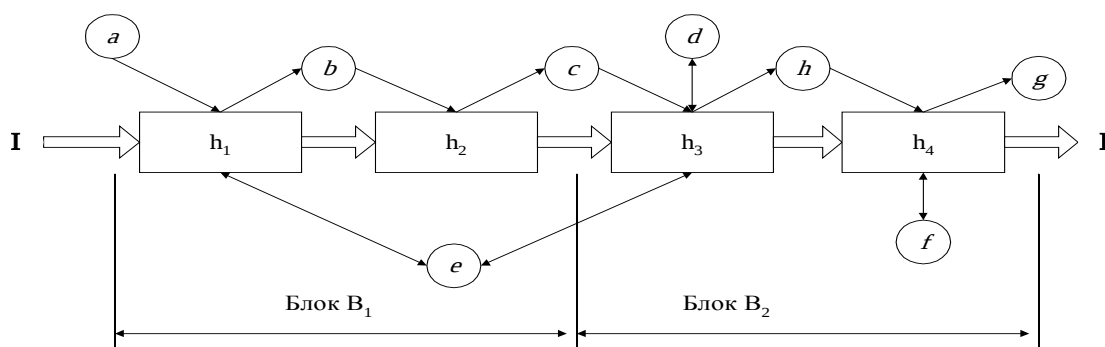
Будем называть трек $\langle h_1, h_2, h_3 \rangle$ вложенным в оператор N_1 . Если сцепление инициатора с элементарным оператором вычисляло одну точку процесса, то сцепление инициатора с обобщенным оператором порождает подпроцесс, равный процессу сцепления инициатора со всеми операторами вложенного трека. Это утверждение следует из требования плотности разбиения. Отсюда следует, что время сцепления инициатора с обобщенным оператором равно времени прохождения инициатора по вложенному треку. Это свойство называют также временной вложенностью. Если вложенный трек свернут в структуру, то, в общем случае, будем говорить о вложенных описаниях. Таким образом, используя рекурсивность разбиений треков, формируется многоуровневая система вложенных описаний, обладающая свойством временной вложенности для всех обобщенных операторов на всех смежных уровнях. Переход к описанию обобщенными блоками соответствует операции свертки процессов.

Использование обобщенных операторов позволяет еще более снизить размерность описания процессов, однако необходимо иметь в виду, что для генерации процесса мы должны уметь выполнить операцию развертки для каждого обобщенного оператора, определив все вложенные треки. В ходе анализа процессов функционирования системы удобно использовать

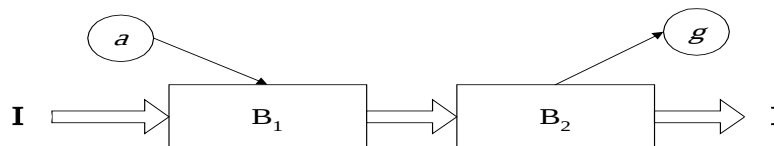
вложенность процессов описаний обобщенных операторов достаточно высокого уровня. Однако в ходе реализации процесса необходимо иметь возможность последовательной развертки обобщенных операторов верхних уровней в треки операторов нижних уровней вплоть до получения треков элементарных операторов.

Совокупность обобщенного оператора и связанных с ним параметров образует *блок*. Какие параметры включаются в состав блока, зависит от постановки задачи моделирования и заранее никак не регламентируется.

На рис.6 приведен пример построения блока. На рис. 6.а. показана операторно-параметрическая схема трека элементарных операторов. На этой схеме выполнено разбиение на два блока B_1 и B_2 . Как видно из рис. 6.в., к блоку B_1 отнесены операторы h_1 и h_2 , а также параметры b, c, e . К блоку B_2 относятся операторы h_3, h_4 и параметры d, h, f (рис. 6.г.). Параметры a и g не принадлежат ни одному из блоков. Полученная блочная схема приведена на рис. 6.б.



а) операторно-параметрическая схема



б) блочная схема

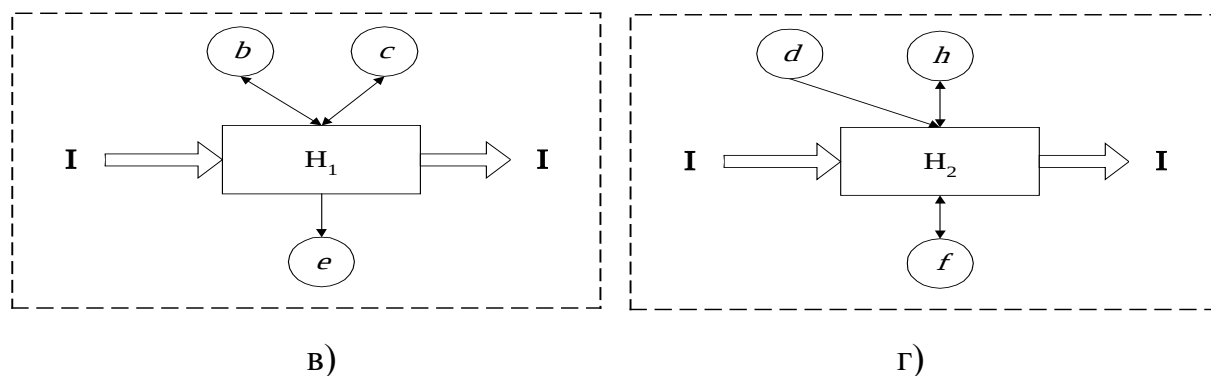


Рисунок 6. К понятию блока

Любой параметр по отношению к заданному блоку может быть внутренним, если включен в состав блока, либо внешним, если не принадлежит блоку. Так, параметр a - внешний по отношению к блоку B_1 . Внутренние параметры, в свою очередь, могут быть входными (параметр d блока B_2), рабочими (параметры h, f блока B_2 и параметр b блока B_1) и выходными (параметр e блока B_1).

Таким образом, можно определить блок общего вида как структуру и связанную с операторами структуры совокупность параметров. Процесс в блоке начинает развиваться, когда в блок поступает инициатор. Принципиально будем различать два типа блоков: *агрегат* и *процессор*.

Агрегат.

На практике часто возникает необходимость описывать процесс функционирования некоторой машины по преобразованию значений параметров в соответствие с заданным циклическим алгоритмом. Такой процесс можно описать с помощью блока общего вида, в котором существует один инициатор, а трек циклически замкнут. Блок, в котором развивается один единственный циклический процесс, будем называть агрегатом, или А-блоком.

Будем в дальнейшем считать, что инициатор находится в агрегате по определению, что избавляет нас от необходимости разработки процедур его доставки в блок. Таким образом, агрегат содержит единственный инициатор

и трек элементарных операторов, замкнутый внутри блока. Поступление каких-либо внешних инициаторов в агрегат невозможно. Будем обозначать агрегат на схемах в виде (рис. 7 а).

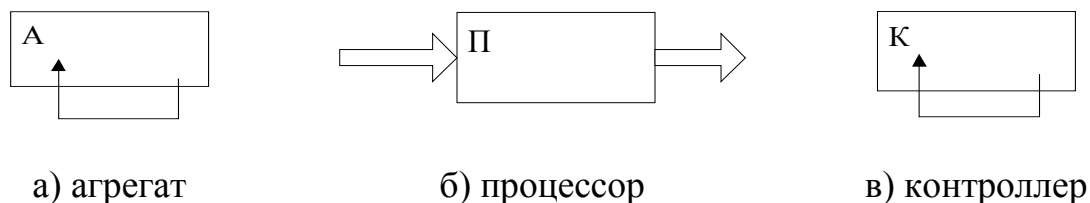


Рисунок 7. Обозначения блоков

Обмен между агрегатом и другими блоками возможен исключительно посредством параметров. Поскольку в агрегате развивается единственный процесс, то не имеет смысла использовать в нем локальную среду и объединенные операторы.

Таким образом, агрегат представляет собой некоторую структуру, порождающую замкнутый трек, и включает единственный внутренний инициатор, находящийся в блоке по определению.

Процессор.

Блок, предназначенный для генерации процессов, инициаторы которых являются внешними по отношению к блоку, называется процессором, или П-блоком. Инициаторы, поступившие извне, сцепляются с блоком, порождая процессы, и затем покидают его. Поскольку процессор генерирует множество одновременно протекающих процессов, в нем используются исключительно объединенные элементарные операторы, а инициаторы должны содержать локальные среды. Таким образом, процессор представляет собой описание произвольной структуры, содержащей объединенные операторы. Процессы порождаются в этом блоке лишь при поступлении в него извне инициаторов, содержащих локальные среды. Из вышесказанного следует, что процессор порождает параллельно протекающие во времени подобные процессы.

Будем обозначать процессор на схемах в виде (рис. 7. б).

Контроллер.

Рассмотрим вновь блок типа агрегат. Как было показано выше, он не имеет возможности взаимодействовать с внешними инициаторами. С тем, чтобы снять это ограничение, введем над инициаторами операции *пассивизации* и *активизации*.

Операция *пассивизации* переводит инициатор в класс обычных параметров.

Операция *активизации*, наоборот, обычный параметр переводит в класс инициаторов.

Если агрегат содержит операторы, выполняющие указанные операции, то такой агрегат назовем контроллером, или К-блоком. Контроллер, таким образом, представляет собой агрегат, выполняющий операции над внешними инициаторами в соответствии с собственным алгоритмом функционирования. Операции над инициаторами суть операции над процессами. Таким образом, контроллер исполняет роль управляющего звена в некоторой блочной схеме. Будем обозначать контроллер на схемах как (рис. 7.в).

Схемы описаний функционирования системы

При описании функционирования системы обычно строится блочная схема, задающая логику взаимодействия, пересечения, уничтожения множества параллельно существующих процессов. Рассмотренная в настоящей статье формализованная модель описания процессов дает возможность с единых позиций провести классификацию существующих подходов в этой области.

Агрегативная схема.

Агрегативная схема предполагает использование только А-блоков. Как показано выше, в такой схеме взаимодействие может осуществляться лишь через параметры. Пример агрегативных моделей рассмотрен в [1], где показано, что каждый А-блок в такой модели может представляться некоторым конечным автоматом.

Если функционирование системы может быть описано совокупностью конечных автоматов, взаимодействующих между собой через множество входных и выходных параметров, то применение агрегативных схем представляется наиболее рациональным. К таким системам можно отнести электронные схемы, блочные схемы функционирования АСУ и т.п.

Процессная схема.

Если в основе описания функционирования системы лежит применение П-блоков, то такое описание называется процессным. Взаимодействие в процессных схемах осуществляется, в основном, через локальные среды процессов. Типичным примером процессного подхода является система имитационного моделирования SIMULA-2, включенная впоследствии в язык SIMULA-67. По существу процесс функционирования системы разбивается на совокупность пересекающихся подобных процессов в объектах. Для каждой такой совокупности формируется свой П-блок.

Процессный подход наиболее эффективен, когда имеем дело с множеством явно выраженных локальных процессов, например, при описании биологических, экономических, социальных и т.п. систем.

Агрегативно-процессные схемы.

Наиболее распространен подход при описании функционирования сложной системы с применением всех видов блоков - А, П и К. Примерами могут быть системы моделирования GPSS, SOL, СЛЕНГ, ДИС и другие. В каждой из них существуют свои ограничения на алгоритмы этих блоков. Так, в языке GPSS существуют операторы-агрегаты (GENERATE, TERMINATE), операторы навигационного типа (TRANSFER), объекты-контроллеры (STORAGE). Большинство остальных операторов имеют тип процессоров. В языке GPSS введены инициаторы в явном виде (TRANSACT), доступные пользователю и способные создавать локальные среды (параметры транзактов). Однако, в языке GPSS пользователь лишен возможности задавать алгоритмы блоков и использует лишь библиотечные конструкции. В языках SOL и СЛЕНГ сделана попытка устранить эти недостатки.

Потоковая схема.

Если в блочной схеме общего вида ограничиваться использованием А-блоков с простыми алгоритмами, а саму схему изобразить в виде сети А-блоков, где линии связи соответствуют движению инициаторов, то получим потоковую схему. Такая схема отражает движение потоков инициаторов между блоками. Поскольку в такой схеме нет процессоров, то инициаторы, с одной стороны, являются носителями последовательности выполнения операторов в своих процессах, а с другой стороны, иницируют процессы в агрегатах. Учитывая, что алгоритмы блоков достаточно просты, то построение потоковых схем может быть выполнено без особых трудностей. Однако необходимо иметь в виду, что при этом существенно возрастает размерность самого описания, нередко теряется обзоримость схемы в целом. Примерами таких схем являются сети Петри, Наура, Е-сети.

Сети массового обслуживания. Сети массового обслуживания являются особым видом потоковых схем, когда все процессы пересечены на ресурсах. Разрешение конфликтных ситуаций в этих схемах выполняется с помощью контроллеров, которые называются системами массового обслуживания (СМО). Система массового обслуживания есть объединение К-блока и одного или нескольких А-блоков: К-блок реализует дисциплину обслуживания требований, а А-блоки - процесс изменения параметров ресурса.

При исследовании систем массового обслуживания аналитическими методами описываются достаточно простые алгоритмы К-блоков, а алгоритмы А-блоков сводятся, как правило, к задержке инициатора на некоторое время в ресурсе. Такие системы могут содержать единственный К-блок, совмещающий вышеуказанные действия.

Таким образом, формализация функционирования систем в виде сети массового обслуживания возникает в том случае, если в схемах общего вида отсутствуют независимые процессоры, а все процессы пересечены на ресурсах. Поток требований на каждый контроллер есть не что иное, как

поток инициаторов процессов, а каждое отдельное требование, будучи инициатором, определяет процесс.

Таким образом, в статье показана конструктивность применения алгоритмической модели описания процессов, позволившая произвести свертку треков описаний, задать однородные процессы, классифицировать блочные структуры описаний, описать и классифицировать различные схемы описания потоковых схем, определить условия применения методов теории массового обслуживания.

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: Дис...док. тех. наук.-М.,2000.-350с.

Integration of algorithmic description of system functioning 77-30569/292913

12, December 2011

Chernen'kii V.M.

Bauman Moscow State Technical University

chernen@bmstu.ru

Article deals with the methods of union the descriptions of discrete system functioning, represented as algorithmic models. The way of process description as an operator-parametric scheme was considered. The concept of homogeneous process was defined. It was shown that the functioning of homogeneous processes could be briefly described with introduction of new united elementary operator and the concepts of local process environments, connected to initiator. Obtained results allowed introducing the concept of blocks and classify them. It, in turn, allowed specifying different approaches to system functioning description; a class of streaming description schemes was defined and analyzed.

Publications with keywords: [blocks](#), [discrete systems](#), [functioning processes](#), [homogeneous processes](#), [local environments](#), [blocking structures](#), [streaming schemes](#), [algorithmic models](#)
Publications with words: [blocks](#), [discrete systems](#), [functioning processes](#), [homogeneous processes](#), [local environments](#), [blocking structures](#), [streaming schemes](#), [algorithmic models](#)

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, 350 p.