

## ОСНОВЫ ПРОЦЕССНОГО ОПИСАНИЯ ФУНКЦИОНИРОВАНИЯ СИСТЕМ

В.М. Черненький, М.В. Черненький (Москва)

Сложная система отличается многочисленными характеристиками, определяемыми аспектами ее описания. Нас будет интересовать аспект описания процесса ее функционирования. Под функционированием системы будем понимать процесс изменения ее состояния во времени. Изложение метода описания опирается на теорию описания параллельно-последовательных процессов, представленную в [3,4,5,6].

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  $\sigma(q_i)$ . Определим пространство состояний системы, как Декартово произведение множеств:

$$S = \prod_{\forall i} \sigma(q_i)$$

Элемент пространства  $S$  есть возможное *состояние системы*.

Используем определение процесса, данное в [1], определим процесс  $Z$  как четверку

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

где  $S$  – пространство состояний системы;  $T$  – множество моментов времени изменения состояний системы;  $F$  – график процесса, определяемый как функциональное отображение  $T \rightarrow S$ ;  $\alpha$  – отношение линейного порядка на  $T$ . В общем случае множества  $T$  и  $S$  могут быть как дискретными, так и непрерывными.

Введем понятие подпроцесса  $Z^i$  как плотного во времени подмножество процесса  $Z$  на интервале  $[t_i; t_j]$ . Плотность по времени означает, что на интервале  $[t_i; t_j]$  нет ни одной точки, принадлежащей  $T$  и не относящейся к подпроцессу  $Z^i$ . Понятие подпроцесса позволяет рассматривать процесс в виде последовательности подпроцессов и производить операции разделения и объединения фрагментов процесса.

Будем рассматривать объект как составную часть системы, характеризующуюся параметрическим множеством объекта  $O_i \subset Q$ . В дальнейшем для краткости, когда это не вызывает двусмысленности, вместо понятий параметрическое множество системы и параметрическое множество объекта будем говорить система и объект соответственно.

Пространство состояний  $S_{O_i}$  объекта  $O_i$  определяется аналогично системе, как

$$S_{O_i} = \prod_{q_i \in O_i} \sigma(q_i).$$

Будем предполагать, что система всегда имеет полное разбиение на объекты. Таким образом:  $\bigcup_{\forall l} O_l = Q$ . Разбиение является непересекающимся, если

$$O_m \cap_{m \neq l} O_l = \emptyset.$$

В противном случае разбиение произведено на пересекающиеся объекты. Если задан процесс  $Z_Q$  в системе, то процесс в объекте  $O_l$  может быть определен как  $Z_{O_l} = \text{Пр}_{S_{O_l}} Z_Q$ , где  $\text{Пр}$  – операция проецирования процесса  $Z_Q$  на пространство объекта  $O_l$ .

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

$$s_{t_i}^{O_l} = H^{O_l}(A^{O_l}, t_i, \omega)$$

где  $t_i \in T_{O_l}$ ;  $A$  – множество аргументов:  $A \subseteq Q$ ;  $\omega$  – случайное число.

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  $O_l \rightarrow O_m$ . Если  $O_m \cap A_{t_i}^{O_l} \neq \emptyset$ , то объект  $O_l$  сцеплен с объектом  $O_m$  в момент времени  $t_i$ :  $O_m \rightarrow O_l$ . Если одновременно  $O_m \cap A_{t_i}^{O_l} \neq \emptyset$  и  $O_l \cap A_{t_i}^{O_m} \neq \emptyset$ , то объекты  $O_m$  и  $O_l$  взаимно-сцеплены в момент времени  $t_i$ :  $O_m \leftrightarrow O_l$ . При операторном способе описания процессов всегда нежелательна модель, приводящая к появлению взаимно – сцепленных объектов, поскольку возникающую неопределенность приходится раскрывать путем решения в общем случае систем нелинейных уравнений, что может привести к непреодолимым трудностям. В дальнейшем будем стремиться создавать модели, не приводящие к взаимному сцеплению объектов.

Следует отметить, что отношение сцепления не является транзитивным. Если к отношению сцепления добавить полное транзитивное замыкание, то получим отношение зависимости.

Конечно, всегда желательно иметь возможность задания процесса в виде единого оператора, однако это, как правило, либо затруднительно, либо невозможно. Введем описание оператора  $H$  как некоторую алгоритмическую структуру.

Рассмотрим дискретный во времени процесс  $Z$ . Пространство состояний  $S$  может быть как непрерывным, так и дискретным. Поставим в соответствие каждой  $i$ -й 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  $t_i$ :  $s_i = h_i^c(A_i, t_i, \omega)$ .

Поскольку оператор  $h_i^c$  описывает вычисление только одной  $i$ -й точки процесса  $Z$ , будем в дальнейшем называть этот оператор *элементарным*. Таким образом, если график процесса содержит  $n$  точек, то мы должны задать линейную последовательность элементарных операторов:

$$h_1^c, h_2^c, \dots, h_i^c, \dots, h_n^c, \text{ называемую в дальнейшем треком TR.}$$

Введем особый элемент модели – инициатор. Первоначально будем полагать, что инициатор – это объект, обладающий следующими фундаментальными свойствами:

- а) независимостью: может существовать самостоятельно без операторов;
- б) динамичностью: инициатор имеет возможность перемещаться от оператора к оператору; будем называть попадание инициатора на оператор *сцеплением* инициатора с элементарным оператором;
- в) инициативностью: в момент сцепления инициатора с оператором происходит выполнение (иницирование) элементарного оператора, что соответствует вычислению нового состояния процесса.

Будем в дальнейшем полагать, что выполнение элементарного оператора происходит мгновенно. Это ограничение не сужает применимости предлагаемой модели, поскольку, если необходимо описать процесс, где вычисление нового состояния требует затрат реального времени, то можно ввести два элементарных оператора, ограничивающих начальный и конечный момент времени интервала вычислений. 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.

Предлагаемая модель описания процесса предполагает, что моменты сцепления инициатора с элементарными операторами определяют сами элементарные операторы. С этой целью введем 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  $h_{i+1}^c$ ;
- в) комбинированная форма, включающая варианты а) и б).

Таким образом:  $h_i^y \in \{h_i^t, h_i^l, h_i^{t,l}\}$

где:  $h_i^y$  – оператор условия сцепления;

$h_i^t$  – оператор временного условия (соответствует варианту а);

$h_i^l$  – оператор логического условия (соответствует варианту б);

$h_i^{t,l}$  – оператор комбинированного условия (соответствует варианту в).

Расширим понятие элементарного оператора, добавив к нему помимо оператора  $h_i^c$  оператор  $h_i^y$ . Таким образом, определим элементарный оператор  $h_i$ , как двойку:

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

Теперь можно определить понятие *алгоритмической модели процесса* (АМП), в виде тройки:

$$\text{АМП} = \langle \{h_i\}_{i=1}^n, \beta, I \rangle,$$

где  $\{h_i\}_{i=1}^n$  – множество элементарных операторов;  $\beta$  – линейный порядок на  $\{h_i\}_{i=1}^n$ ;  $I$  – инициатор.

Следует обратить внимание, что АМП содержит один и только один инициатор, т.е. каждому процессу соответствует один инициатор. В этом смысле инициатор является представителем процесса, при его потере либо отсутствии развитие процесса прекращает-

ся. Если в системе развивается одновременно  $m$  процессов, то в модели присутствует  $m$  инициаторов.

Таким образом, трек можно определить как  $TR = \langle \{h_i\}_{i=1}^n, \beta \rangle$ .

Процесс задан, если задан трек элементарных операторов и инициатор.

Рассмотрим некоторый трек элементарных операторов, выполняющих одни и те же операции над аргументами. Известно, что операторы эквивалентны, если при одних и тех же значениях аргументов они вычисляют одинаковые результаты. Введение понятия эквивалентных операторов позволяет задать отношение эквивалентности на множестве

$\{h_j\}_{j=1}^n$  элементарных операторов трека  $TR$ .

Структурой назовем свертку трека  $TR$  по отношению эквивалентности элементарных операторов. Очевидно, если заданы трек и отношение эквивалентности операторов, то всегда возможно построение структуры. Однако обратное восстановление трека по структуре является неоднозначной операцией. С тем, чтобы операцию построения трека из структуры сделать однозначной, введем еще один тип элементарного оператора – навигационный элементарный оператор. Навигационный оператор определяется так же, как и элементарный оператор, однако в результате его выполнения определяется тот элементарный оператор в структуре, который должен выполняться следующим. Выполнение навигационного оператора определяется инициатором. Поскольку время на выполнение навигационного оператора, как и всех элементарных операторов, равно нулю, то использование его не сказывается на времени реализации процесса. В общем случае навигационный оператор должен следовать за каждым элементарным оператором в структуре.

Если на треке операторов указать используемые этими операторами параметры и их взаимосвязи, то получим операторно-параметрическую схему, дающую достаточно полное представление о способе описания процесса с использованием АМП.

Рассмотрим случай задания двух процессов  $Z_1$  и  $Z_2$ , у которых одинаковый трек, но различные параметры, взаимодействующие с элементарными операторами. Операторно-параметрические схемы таких процессов близки друг к другу. С тем, чтобы объединить описания процессов, 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. Назовем эту совокупность локальной средой процесса. В дальнейшем будет удобно 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. Одинаковые операторы в треках объединяются в один объединенный оператор, который отличается от простых возможностью взаимодействовать не только с явно заданными параметрами, но и с параметрами локальных сред инициаторов.

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

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. Плотное разбиение может быть выполнено

и на треке обобщенных операторов. Рекурсивность определения позволяет вводить понятия обобщенных операторов различного уровня: первого, второго и т.д.

Совокупность обобщенного оператора и связанных с ним параметров образует блок. 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*.

Агрегатом, или А-блоком, назовем некоторую структуру, порождающую замкнутый трек, и включающую единственный внутренний инициатор, находящийся в блоке по определению.

Блок, предназначенный для генерации процессов, инициаторы которых являются внешними по отношению к блоку, назовем 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. Из вышесказанного следует, что процессор порождает параллельно протекающие во времени подобные процессы.

Введем в рассмотрение еще один тип блоков – *контроллер*. Структурно контроллер построен как агрегат. 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 параметров типа ссылки (на локальную среду). Операция активизации переводит параметр ссылки 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.

Процессы  $Z_i$  в системе  $Q$  развиваются параллельно. Это значит, что они изменяют значения параметров системы в течение одного и того же интервала времени. Достаточно типичны ситуации, когда по логике функционирования системы накладываются ограничения на изменение некоторых параметров несколькими процессами одновременно в течение заданного либо обусловленного интервала времени.

Совокупность параметров системы, на изменение которых сформулированы некоторые ограничивающие условия, назовем ресурсом  $R$ . Таким образом,  $R \subset Q$ . Если объект  $O_k$  изменяет параметры ресурса  $R$ , то  $R \subset O_k$ . Захват ресурса  $R$  процессом  $Z$  означает получение разрешения процессу  $Z$  изменять значения параметров  $q \in R$ .

Конфликт на ресурсе есть возникновение ситуации, когда тому или иному процессу отказано в захвате ресурса до момента выполнения некоторого наперед заданного условия. Из определения ресурса следует, что конфликт на ресурсе возможен лишь для пересекающихся объектов. Таким образом, необходимо добиться согласования процессов в этих объектах.

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

Агрегативная схема предполагает использование только *A*-блоков, в которой взаимодействие может осуществляться лишь через параметры. Пример агрегативных моделей рассмотрен в [7], где показано, что каждый *A*-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.

Если в основе описания функционирования системы лежит применение *П*-блоков, то такое описание называется процессным. Взаимодействие в процессных схемах осуществляется через параметры и через локальные среды процессов. 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.

На практике при описании функционирования систем более распространены агрегативно-процессные схемы с применением всех видов блоков – *A*-, *П*- и *К*-.

Если в блочной схеме общего вида ограничиваться использованием блоков с простыми алгоритмами, а саму схему изобразить в виде сети *A*-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, с которыми они сцепляются. 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 обзорность схемы в целом.

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. Система массового обслуживания есть объединение *К*-блока, реализующего дисциплину обслуживания требований, и одного или нескольких *A*-блоков, определяющих процесс изменения параметров ресурса. При исследовании систем массового обслуживания аналитическими методами выбираются достаточно простые алгоритмы *К*-блоков, а алгоритмы *A*-блоков сводятся, как правило, к задержке инициатора на некоторое время в ресурсе.

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.

### Литература

1. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. – М.: Сов.Радио, 1973. – 438 с.
2. Дейкстра Э. Взаимодействие последовательных процессов. – М.: Мир, 1972. – 364 с.
3. Черненький В.М. Процессно-ориентированная концепция системного моделирования АСУ: дис. ... докт. техн. наук. – М., 2000. – 299 с.

4. **Черненький В.М.** Интеграция алгоритмических описаний функционирования системы // Наука и Образование: электронное научно-техническое изд. – 2011. – Вып. 12. [<http://technomag.edu.ru/doc/292913.html>].
5. **Черненький В.М.** Алгоритмическая модель описания дискретного процесса функционирования системы // Наука и Образование: электронное научно-техническое изд. – 2011. – Вып. 12. [<http://technomag.edu.ru/doc/292620.html>].
6. **Черненький В.М.** Понятие и свойства дискретного процесса функционирования системы // Наука и Образование: электронное научно-техническое изд. – 2011. – Вып. 12. [<http://technomag.edu.ru/doc/292510.html>].
7. **Бусленко В.Н.** Автоматизация имитационного моделирования сложных систем. – М.: Наука, 1977. – 239 с.