ЭМПИРИЧЕСКОЕ ПРЕДСКАЗАНИЕ И РАСПОЗНАВАНИЕ ОБРАЗОВ
(Вычислительные системы)
1979 год Выпуск 79

УДК 519.682.6

С Т И М О Л – ЯЗЫК ПРЕДСТАВЛЕНИЯ СТРУКТУРНЫХ ИМИТАЦИОННЫХ МОДЕЛЕЙ

Н.И.Дубовская, Д.И.Свириденко

Основные принципы построения имитационных моделей данного класса были обсуждены в [1], поэтому настоящую работу следует рассматривать как ее продолжение.

Язык моделирования является достаточно мощным и гибким инструментом в руках исследователя лишь в том случае, если он содержит развитый аппарат общеалгоритмических средств. В связи с этим в качестве подмножества языка СТИМОЛ выбран язык PL/I, синтезирующий в себе ряд лучших идей и понятий языков АЛГОЛ, ФОРТРАН и КОБОЛ и, кроме того, обладающий такими новыми возможностями, как оперирование с широким набором типов данных, создание из них сложных структур, организация программной реакции на прерывания, возможность параллельных вычислений в реальном масштабе времени за счет одновременного использования различных физических устройств ЭВМ, гибкость средств размещения памяти и ряд других.

Расширение языка PL/I заключается во введении набора средств – ряда новых описаний, операторов и стандартных процедур, предназначенных для моделирования параллельно протекающих и взаимодействующих между собой процессов сложных систем.

Язык ориентирован на моделирование систем на машинах серии ЕС ЭВМ в ОС/ЕС.

Программная модель, составленная на языке СТИМОЛ (в дальнейшем называемая СТИМОЛ-программой), представляет собой совокупность описаний процессов и требуемого числа процедур, используемых в процессе моделирования. Одна из них, обозначенная как главная и содержащая в списке своих утверждений описание структуры модельной дискретной сети и оператор инициирования функци-

онирования сети - оператор запуска МДС, определяет начало выполнения программы.

Исполнение СТИМОЛ-программы состоит из последовательного выполнения активных фаз процессов. Установление строгой их очередности производится с помощью специальных системных средств системы управления МДС, моделирующей параллельно в модельном времени протекание отдельных процессов и осуществляющей выполнение модельной дискретной сети сверху вниз.

Ниже приведены описания и операторы вводимых языковых средств и их семантика. Используемый в языке набор функций и процедур для генерирования случайных величин с различными законами распределения, а также обработка результатов экспериментов в данной работе не рассматривается.

Семантика части операторов языка подобна семантике управляющих и планирующих операторов в [2].

I. Формализм описания синтаксиса

В соответствии с традиционными формами Бэкуса-Наура синтаксические конструкции обозначаются словами, заключенными в угловые скобки " ⟨ " и " ⟩ ". Метасимвол ":=" читается "по определению есть". Вертикальная черта есть знак альтернативы. Три точки "..." означают появление непосредственно предшествующего элемента один или несколько раз подряд. Фигурные скобки "{" и "}" используются для объединения нескольких элементов правой части синтаксического определения в один элемент. Квадратные скобки указывают, что заключенные в них элементы либо присутствуют, либо их можно опустить. Внутри скобок может использоваться знак альтернативы. К совокупности элементов, заключенных в скобки, может применяться знак повторения.

2. Описания

2.I. Описание процесса.

⟨описание процесса⟩ ::= ⟨заголовок процесса⟩ ⟨тело процесса⟩ ⟨заключительный оператор⟩

⟨заголовок процесса⟩ ::= ПРОЦЕСС ⟨имя процесса⟩ [(⟨список формальных параметров⟩ [1])] ;

I) Данная конструкция задается так же, как список параметров процедуры. Отмеченные символом "•" синтаксические конструкции можно найти в [3].

⟨ имя процесса ⟩::=⟨ идентификатор⟩*

⟨ тело процесса⟩ ::=⟨ утверждение⟩ ...

⟨ утверждение⟩ ::=⟨ P-процедура ⟩|⟨ P-оператор⟩|⟨ R -оператор⟩

⟨ P-процедура⟩ ::=⟨ PROCEDURE-утверждение⟩*{⟨ P -оператор⟩|⟨P-процедура⟩} ...⟨ END -утверждение⟩*

⟨ P-оператор ⟩::= ⟨ PL-оператор ⟩*1)|⟨ L-оператор ⟩|{⟨ BEGIN-утверждение ⟩*|⟨ DO-утверждение ⟩*}{⟨ P-оператор ⟩|⟨ P-процедура ⟩}... ⟨ END-утверждение ⟩*| IF ⟨ элементное выражение ⟩*THEN ⟨ P-оператор ⟩[ELSE⟨ P-оператор⟩]

⟨ R-оператор⟩ ::=[⟨ метка ⟩* :]...{⟨ оператор задержки ⟩ |⟨ δ R-оператор ⟩|⟨ оператор непосредственного планирования ⟩|⟨ оператор останова текущего процесса ⟩}

⟨ L-оператор⟩ ::=[⟨ метка ⟩*:]...{ ⟨δ -оператор ⟩|⟨ оператор активизации подсети МДС ⟩|⟨ оператор запуска по времени ⟩|⟨ оператор запуска по условию ⟩|⟨ оператор запуска с задержкой ⟩|⟨оператор завершения текущего процесса ⟩|⟨ оператор завершения внешнего процесса ⟩|⟨ оператор останова внешнего процесса ⟩ | ⟨ оператор трансформации сети ⟩ | ⟨ оператор задания очереди ⟩ | ⟨ оператор прекращения моделирования ⟩}

⟨ заключительный оператор ⟩ ::=ЗАВЕРШИТЬ;

Описание процесса в СТИМОЛ-программе представлено в форме процедуры и состоит из трех частей: заголовка,тела и заключительного оператора.

Заголовок определяет название процесса и набор его формальных параметров.

Тело процесса есть некоторая последовательность утверждений, определяющих структуру данных и действия процесса.

Заключительный оператор является оператором завершения программы процесса.

Как и PL-процедура, процесс может иметь произвольную блочную структуру. В группу R-операторов выделены операторы СТИМОЛ-язы -

---

I) В группу PL-операторов объединены все PL-утверждения [3] (включая утверждение DECLARE), за исключением утверждений BEGIN и PROCEDURE, являющихся заголовками блоков, утверждения DO , озаглавливающего группу, а также составного утверждения IF. Использование в теле блоков, в DO-группах и в утверждении IF PL-операторов соответствует правилам [3].

ка, назначающие процессу, в теле которого они используются, некоторую вторичную точку входа в его программу. В связи с этим не допускается их использование во вложенных в программу процесса блоках, в DO-группах и утверждении IF . Все остальные операторы СТИМОЛ-языка, за исключением оператора запуска сети, не используемого в теле процессов, объединены в группу L-операторов.

Любые переменные, декларированные локально в теле процесса, т.е. определенные PL-оператором DECLARE с описателем INTERNAL , являются атрибутами этого процесса. Атрибуты, значения которых должны быть сохранены при исполнении операторов, прерывающих работу процесса, должны иметь описатель STATIC. Общие переменные для программ различных процессов задаются с описателем EXTERNAL .

Выполнение программы процесса, активизируемого системой управления МДС, начинается с соответствующей определенной для него точки реактивации. В начале моделирования для каждого процесса устанавливается точка реактивации, соответствующая первому оператору его программы.

2.2. О п и с а н и е с т р у к т у р ы МДС.

⟨ описание структуры МДС ⟩ ::= СЕТЬ ⟨ отношение ⟩ [& ⟨ отношение ⟩] ...;

⟨ отношение ⟩ ::= (⟨ имя процесса ⟩ : ⟨ имя процесса ⟩ [, ⟨ имя процесса ⟩] ...)

Описание структуры модели задается конъюнкцией (n-1)-го отношения, где n - число процессов, определенных в модели данным описанием в качестве сетевых. Каждое отношение задает все λ-связи некоторого процесса, имя которого указано слева от двоеточия, с процессами других уровней. Имена последних перечисляются через запятую справа от двоеточия. Исходный сетевой процесс (процесс, находящийся на верхнем уровне) должен быть указан в первом отношении.

Согласно заданному описанию, с помощью системы управления осуществляется выполнение МДС сверху вниз.

3. Операторы

Некоторые используемые в ряде операторов конструкции представляют собой:

⟨ T ⟩ ::= ⟨ арифметическое выражение ⟩*

⟨ Y ⟩ ::= ⟨ булевское выражение ⟩*

⟨X⟩ ::= ⟨ имя процесса ⟩[(⟨ список фактических параметров ⟩ *I))]

Связь между элементами списка фактических параметров операторов СТИМОЛ-языка и элементами списка формальных параметров процессов такая же, как между аргументами вызывающего утверждения PL-языка и параметрами вызываемой процедуры.

3.I. О п е р а т о р ы з а д е р ж к и. Действия операторов задержки распространяются на текущий процесс, вызывая задержку исполнения программы процесса до тех пор, пока не наступит заданный момент модельного времени или не выполнится условие, указанное в операторе. Прерванному процессу назначается точка возврата (точка реактивации) непосредственно за оператором задержки, начиная с которой будет вновь продолжено его функционирование.

3.I.I. О п е р а т о р в р е м е н н о й з а д е р ж к и.
⟨ оператор временной задержки ⟩ ::=(ЗАДЕРЖКА ⟨T⟩| ЖДАТЬ ⟨T⟩) ;

Первым оператором задается интервал времени, через который должна наступить очередная активная фаза текущего процесса; вторым оператором задается непосредственно момент времени ее наступления.

3.I.2. О п е р а т о р у с л о в н о й и л и у с л о в н о - в р е м е н н о й з а д е р ж к и.
⟨ оператор условной или условно-временной задержки ⟩ ::= ЖДАТЬ_ПОКА ⟨Y⟩[{ ИЛИ|ИЛИ_ДО}⟨T⟩];

Конструкция, заключенная в квадратные скобки, может быть пустой. В этом случае оператор прерывает исполнение текущего процесса до тех пор, пока не выполнится заданное условие. В противном случае наступление следующей активной фазы для прерванного процесса будет определяться моментом модельного времени, при котором либо выполнится поставленное условие, либо будет достигнуто значение модельного времени, равное некоторой отметке t . Последняя определяется при использовании конструкции "ИЛИ ⟨T⟩" как смещение относительно текущего модельного времени на интервал, задаваемый значением арифметического выражения T ; при использовании конструкции "ИЛИ_ДО ⟨T⟩" - непосредственно значением выражения T.

3.2. О п е р а т о р ы и н и ц и и р о в а н и я о б р а т н ы х с в я з е й. Определены два типа операторов, вызывающих активизацию сетевых процессов по контурам обратных связей.

I) Данная конструкция задается так же, как список аргументов в CALL-утверждении.

3.2.1. δ - о п е р а т о р.

⟨ δ-оператор ⟩ ::={СТАРТ|ПУСК }⟨ список активизируемых процес - сов ⟩ ;

⟨список активизируемых процессов ⟩ ::= ⟨X⟩[& ⟨X⟩]...

Выполнение активного процесса прерывается. Планируется ис - полнение указанных в операторе процессов на текущий момент модельного времени. При этом оператор старта определяет исполнение каждого активизируемого сетевого процесса,начиная с его первого оператора, а оператор пуска - с соответствующей точки реакти - вации, определенной для данного процесса.

Прерываемому процессу назначается точка реактивации, совпадающая с началом программы его действий. При новом инициировании прерванного процесса будет вновь начато исполнение его программы.

3.2.2. δR - о п е р а т о р.

⟨ δR-оператор ⟩ ::={СТАРТ_R | ПУСК_R }⟨ список активизируемых процессов ⟩ ;

Семантика δR-операторов аналогична семантике группы δ-операторов с той лишь разницей, что прерываемому процессу назначается точка реактивации непосредственно за данным оператором. При новом инициировании прерванного процесса будет продолжено исполнение его программы.

3.3. О п е р а т о р а к т и в а ц и и п о д с е т и МДС.

⟨оператор активизации подсети МДС ⟩:: НАЧАТЬ ⟨X⟩[& ⟨X⟩] ... ;

Оператор используется в теле конечного процесса подсети.Первым указывается имя ее исходного процесса. Затем перечисляются сетевые процессы, принадлежащие данной подсети и декларированные с параметрами. Соответственно для каждого из данных процессов задается список фактических параметров. Процессы, декларированные без параметров, могут не указываться.

Оператор прерывает исполнение активного (конечного) процесса подсети, планируя для ее исходного процесса повторное выполнение его программы на текущий момент модельного времени. Прерванному процессу назначается точка реактивации, совпадающая с первым оператором его программы.

Выполнение программ процессов, принадлежащих данной подсети и последовательно активизируемых средствами системы управления, будет осуществляться, начиная с точек реактивации, определенных для данных процессов. Окончание функционирования подсети соответствует активизации ее конечного процесса.

3.4. О п е р а т о р ы з а п у с к а.С помощью данных операторов осуществляется запуск свободных процессов. Определены четыре типа операторов - запуск по времени, по условию, с задержкой и непосредственного планирования. Все операторы, кроме последнего, не прерывают исполнения текущего процесса и определяют собой планирование "будущих" событий для процессов, указанных в операторах. Исполнение программ процессов будет осуществляться с соответствующих определенных для них точек реактивации.

3.4.1. О п е р а т о р з а п у с к а п о в р е м е н и.
⟨оператор запуска по времени⟩::= ЗАПУСТИТЬ ⟨X⟩ В ⟨T⟩;

Значение выражения T определяет момент модельного времени,на который должна быть запланирована очередная активная фаза процесса X .

3.4.2. О п е р а т о р з а п у с к а п о у с л о в и ю.
⟨оператор запуска по условию⟩::= ЗАПУСТИТЬ ⟨X⟩ ПРИ ⟨Y⟩;

Наступление очередной активной фазы процесса X определяется моментом модельного времени, при котором выполнится заданное условие.

3.4.3. О п е р а т о р з а п у с к а с з а д е р ж к о й.
⟨оператор запуска с задержкой⟩ ::= ЗАПУСТИТЬ ⟨X⟩ ЧЕРЕЗ ⟨T⟩;

Значение выражения T определяет задержку наступления очередной активной фазы процесса X относительно текущего модельного времени.

3.4.4. О п е р а т о р н е п о с р е д с т в е н н о г о п л а н и р о в а н и я.
⟨ оператор непосредственного планирования⟩ ::= ЗАПУСТИТЬ ⟨X⟩;

Оператор прерывает исполнение активного процесса, планируя для процесса X выполнение его программы на текущий момент модельного времени. Вслед за этим порождается уведомление о прерванном процессе. Последнему назначается точка реактивации непосредственно за данным оператором, начиная с которой при том же значении модельного времени будет продолжено его выполнение.

Все описанные операторы запуска выполняются только в том случае, если для указанного процесса ранее не было запланировано наступление его очередной активной фазы. В противном случае оператор не выполняется.

3.5. О п е р а т о р ы п р е к р а щ е н и я и с п о л н е н и я п р о ц е с с а. Операторы данной группы позволяют в

любом месте прекратить исполнение программ процессов и назначить различные точки последующего входа в их программы.

В дальнейшем процесс, выполнение которого было прекращено, может быть вновь активизирован (сетевой - оператором инициирования обратных связей или оператором активизации подсети; свободный - с помощью операторов запуска) и продолжен либо начат заново.

3.5.I. Оператор останова текущего процесса.

⟨ оператор останова текущего процесса ⟩::= ОСТАНОВИТЬ ;

Действие оператора распространяется на текущий процесс, выполнение которого прекращается. Новая активная фаза процесса не планируется. Процессу назначается точка реактивации непосредственно за данным оператором.

3.5.2. Оператор останова внешнего процесса.

⟨ оператор останова внешнего процесса ⟩::= ОСТАНОВИТЬ ⟨ имя процесса ⟩;

Прекращается исполнение процесса, имя которого указано в операторе. Запланированная его очередная активная фаза отменяется. При этом оператор не изменяет значения определенной для него точки реактивации.

Выполнение текущего процесса не прерывается.

3.5.3. Оператор завершения текущего процесса.

⟨ оператор завершения текущего процесса ⟩::= ЗАВЕРШИТЬ;

Семантика оператора аналогична семантике оператора п.3.5.I с той лишь разницей, что процессу назначается точка реактивации, соответствующая первому оператору его программы.

Оператор ЗАВЕРШИТЬ является также обязательным заключительным оператором программ процессов.

3.5.4.Оператор завершения внешнего процесса.

⟨ оператор завершения внешнего процесса ⟩::= ЗАВЕРШИТЬ ⟨ имя процесса ⟩;

Семантика оператора аналогична семантике оператора п.3.5.2 с той лишь разницей, что процессу назначается точка реактивации, соответствующая первому оператору его программы.

3.6. О п е р а т о р ы т р а н с ф о р м а ц и и с е т и.

⟨ оператор трансформации сети ⟩::= {ИСКЛЮЧИТЬ | ВКЛЮЧИТЬ }⟨ отношение ⟩[& ⟨ отношение ⟩]...;

⟨ отношение ⟩::= (⟨ имя процесса ⟩:⟨ имя процесса ⟩[,⟨ имя процесса ⟩]...)

Каждое отношение задает соответствие исключаемые либо включаемые в структуру МДС λ-связи некоторого процесса, указанного слева от двоеточия, с процессами других уровней. Имена последних указываются справа от двоеточия. Выполнение МДС будет продолжаться в соответствии с ее новой структурой.

Исполнение текущего процесса не прерывается.

3.7. О п е р а т о р з а д а н и я о ч е р е д и.

⟨оператор задания очереди⟩ ::= ОЧЕРЕДЬ (⟨имя процесса⟩ {, ⟨имя процесса ⟩}...)

Оператор используется в процессе, имеющем более одной входящей δ-связи, в случае необходимости задания порядка обслуживания процессов. Последние перечисляются в соответствии с требуемой очередностью их обслуживания.

3.8. О п е р а т о р з а п у с к а МДС.

⟨ оператор запуска МДС ⟩::= НАЧАТЬ СЕТЬ [⟨x⟩[,⟨x⟩] ...];

Оператор используется в главной процедуре СТИМОЛ-программы.

Конструкция, заключенная в скобки, представляет собой список сетевых процессов, декларированных с параметрами, с указанием соответствующих им фактических параметров. Задаваемые фактические параметры должны быть определены в главной процедуре описателями того же типа, что и формальные параметры названных процессов.

Выполнением оператора начинается функционирование МДС. Устанавливаются начальные значения указателей точек реактивации для каждого из процессов, соответствующие первым операторам их программ. Устанавливается нулевое значение модельного времени. Активизируется исходный сетевой процесс.

При завершении выполнения конечного процесса МДС управление передается оператору главной процедуры, находящемуся непосредственно за оператором запуска сети.

3.9. О п е р а т о р п р е к р а щ е н и я м о д е л и р о в а н и я.

⟨ оператор прекращения моделирования ⟩::= СТОП;

Оператор может быть использован в программе любого процесса. Выполнение оператора вызывает прекращение функционирования МДС. Управление передается оператору главной процедуры, следующему за оператором запуска сети.

4. СТИМОЛ-программа

⟨СТИМОЛ-программа⟩ ::= СТИМОЛ-НАЧАЛО ⟨описание главной процедуры⟩⟨описание процесса⟩ ...[⟨описание процедуры⟩*] ...
СТИМОЛ-КОНЕЦ

⟨описание главной процедуры⟩ ::= ⟨ PROCEDURE OPTIONS(MAIN) – утверждение⟩* ⟨описание структуры МДС⟩ { ⟨оператор запуска МДС ⟩| ⟨PL-утверждение⟩*}... ⟨ END-утверждение⟩*

СТИМОЛ-программа включает в себя:

– главную процедуру, описываемую так же, как главная процедура PL-программы, список используемых утверждений которой расширен описанием структуры модельной дискретной сети и оператором запуска сети;

– совокупность описаний процессов;

– требуемое число процедур, выражаемых обычными средствами языка PL/I .

Вложение описаний процессов друг в друга не разрешается; не допускается такие описание процессов внутри процедурных или начальных блоков. Описание каждого процесса представляет собой некоторый внешний блок СТИМОЛ-программы.

Описание СТИМОЛ-программы начинается оператором СТИМОЛ-НАЧАЛО и заканчивается оператором СТИМОЛ-КОНЕЦ.

5. Функционирование модели

Выполнение программы моделирования начинается с ее главной процедуры. Исполнение оператора запуска МДС вызывает обращение к системе управления, осуществляющей действия по организации очередности протекания активных фаз отдельных процессов и пересчету модельного времени.

Первым активизируемым процессом является исходный сетевой процесс, определяющий начало функционирования МДС.

При окончании активной фазы текущего процесса функционирование МДС продолжается инициированием очередного процесса из числа готовых для исполнения процессов. Таковыми являются:

1. Сетевые процессы, активизируемые операторами инициирования обратных связей.

2. Исходные сетевые процессы активизируемых подсетей.

3. Свободные процессы, активизируемые операторами непосредственного планирования, и процессы, исполнение которых прервано данным оператором.

4. Процессы, для которых выполнилось указанное условие их активизации.

5. Процессы, для которых значение временной отметки наступления их активных фаз совпадает со значением текущего модельного времени.

6. Сетевые процессы, для которых завершились программы всех процессов, активизируемых ими по контурам обратных связей.

7. Сетевые процессы, для которых оказались "возбужденными" все связи, составляющие конъюнктивные группы λ-входов данных процессов.

Уведомление о готовности процессов первых трех групп порождается при выполнении соответствующих операторов. Готовность к исполнению процессов остальных перечисленных групп проверяется системой управления после завершения каждого события. Проверка и порождение соответствующих уведомлений осуществляется в порядке, указанном выше для данных групп процессов.

Выбор из числа готовых к исполнению процессов активизируемого процесса осуществляется в порядке поступивших о них уведомлений. Если исполнение некоторого сетевого процесса требуется одновременно нескольким процессам других нижележащих уровней и при этом была задана очередность их обслуживания, то соответственно выбор будет осуществляться в указанном порядке.

Если при текущем значении модельного времени ни один из процессов не находится в состоянии готовности, то новым значением модельного времени становится значение ближайшей временной отметки ожидающих процессов.

Окончание функционирования МДС соответствует завершению исполнения конечного сетевого процесса или выполнению оператора СТОП. Управление при этом передается главной процедуре СТИМОЛ-программы, выполнение которой продолжается с оператора, следующего за оператором запуска МДС. Окончание исполнения программы моделирования соответствует выполнению оператора END главной процедуры.

В приложении синтаксис языка СТИМОЛ приведен в графической форме.

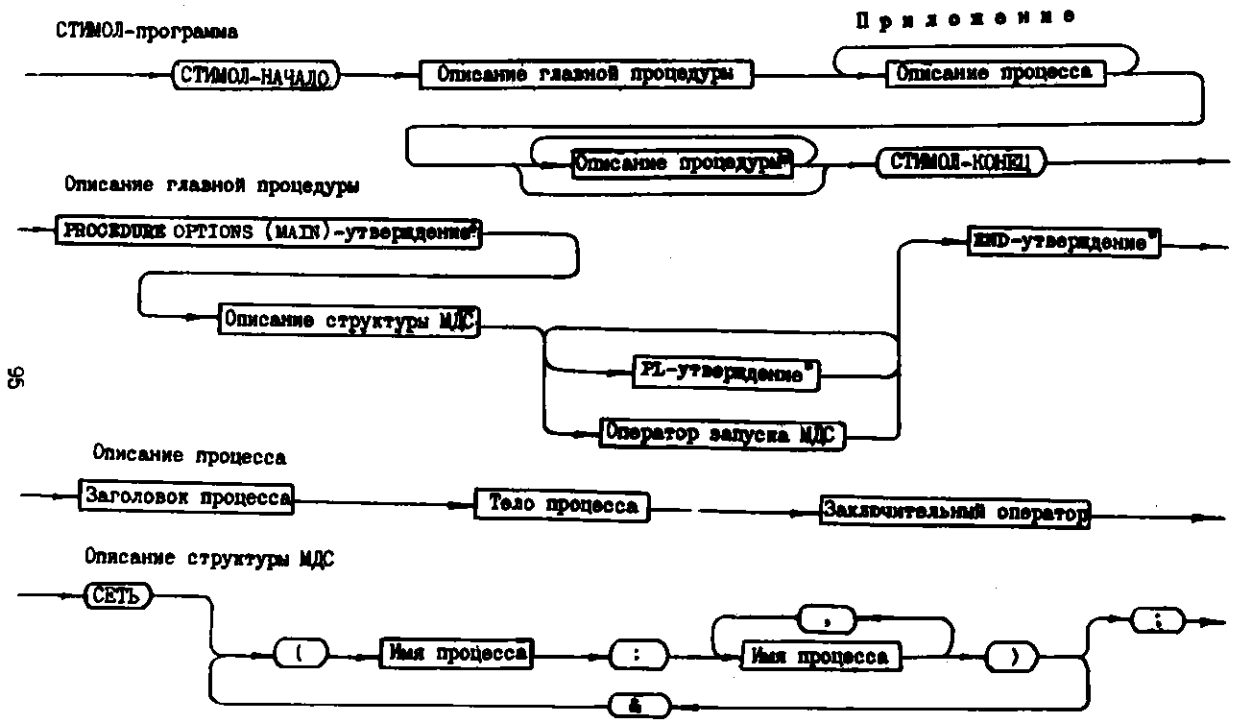
Литература

1. ДУБОВСКАЯ Н.И., СВИРИДЕНКО Д.И. Структурные имитационные модели. -Настоящий сборник, с. 72-81.

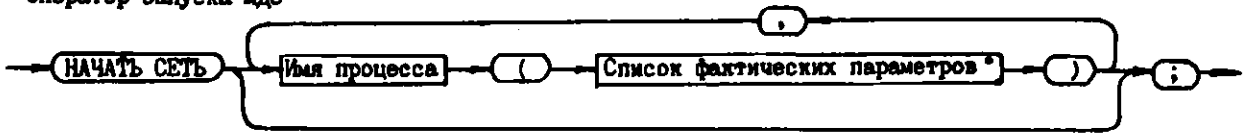
2. ДАЛ У., НИГАРД К. СИМУЛА - язык для программирования и описания систем с дискретными событиями. -В кн.: Алгоритмы и алгоритмические языки. Вып.2. М., 1967, -72 с. (ВЦ АН СССР).

3. Единая система электронных вычислительных машин. Операционная система. ПЛ/I. Описание языка. -Б.м., 1977.

Поступила в ред.-изд.отд.
6 июля 1979 года

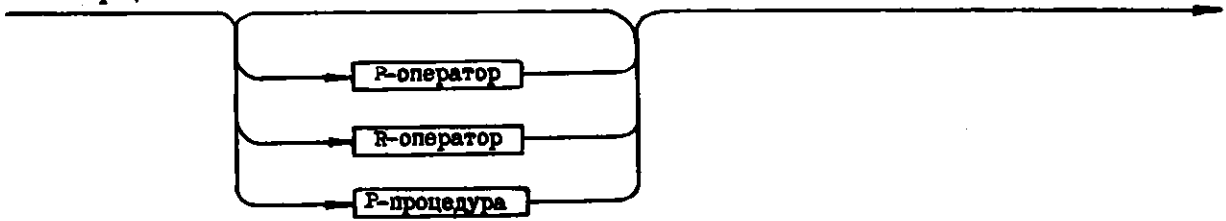

СТИМОЛ-программа
СТИМОЛ-НАЧАЛО
Описание главной процедуры
Приложение
Описание процесса
Описание процедуры
СТИМОЛ-КОНЕЦ
Описание главной процедуры
PROCEDURE OPTIONS (MAIN)-утверждение
Описание структуры МДС
PL-утверждение
Оператор запуска МДС
END-утверждение
Описание процесса
Заголовок процесса
Тело процесса
Заключительный оператор
Описание структуры МДС
СЕТЬ
(
Имя процесса
:
,
Имя процесса
)
;
&

Оператор запуска МДС

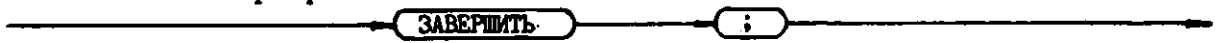


Заголовок процесса

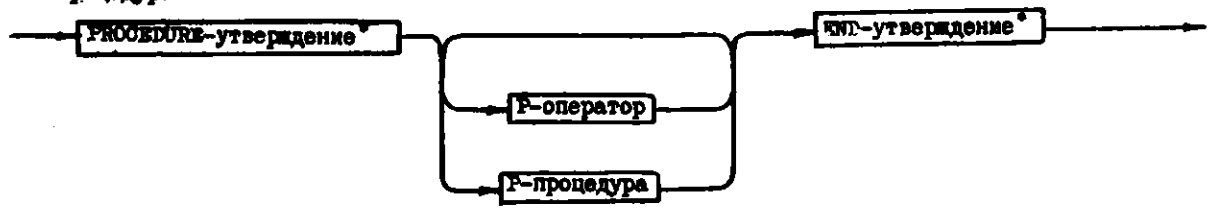
Тело процесса



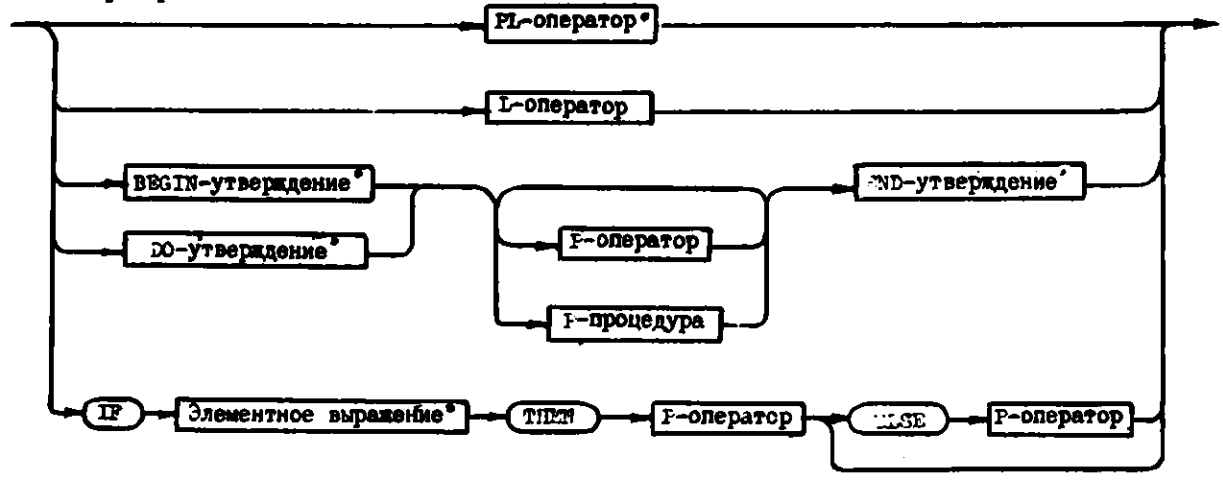
Заключительный оператор

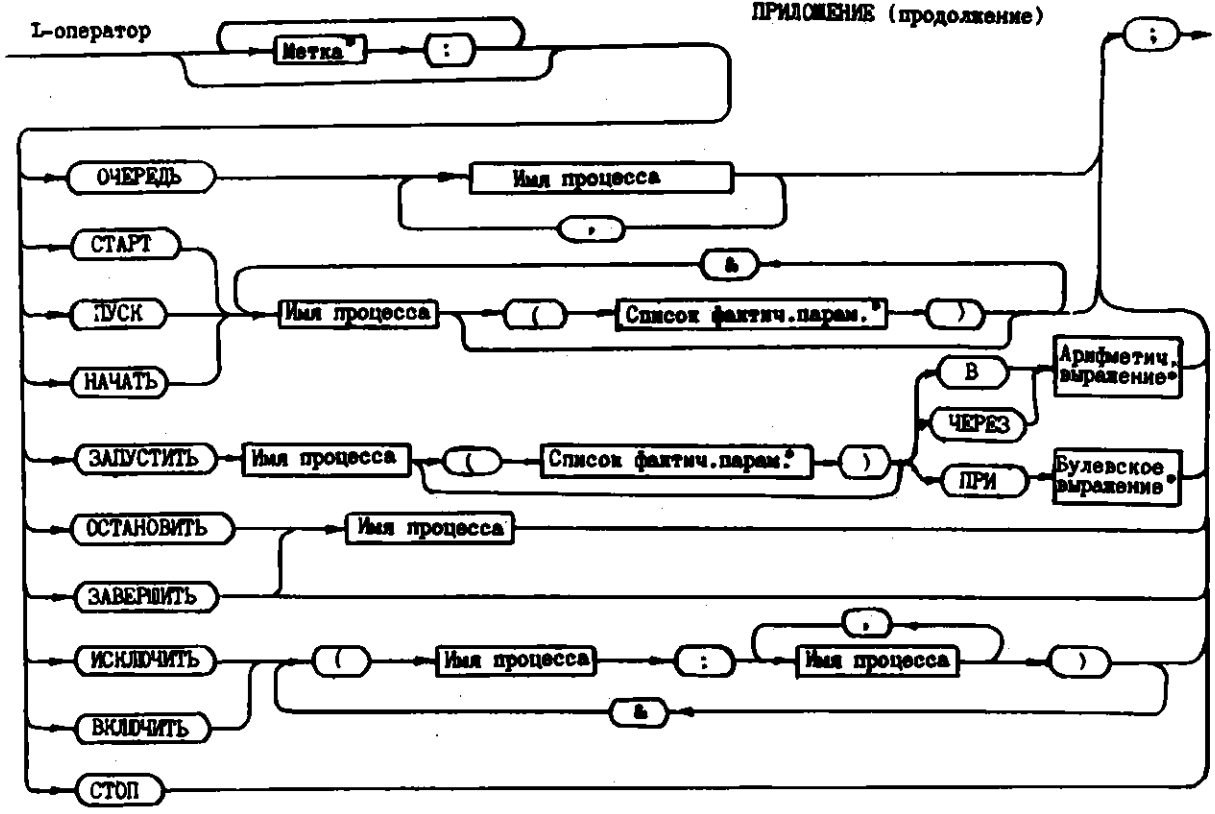


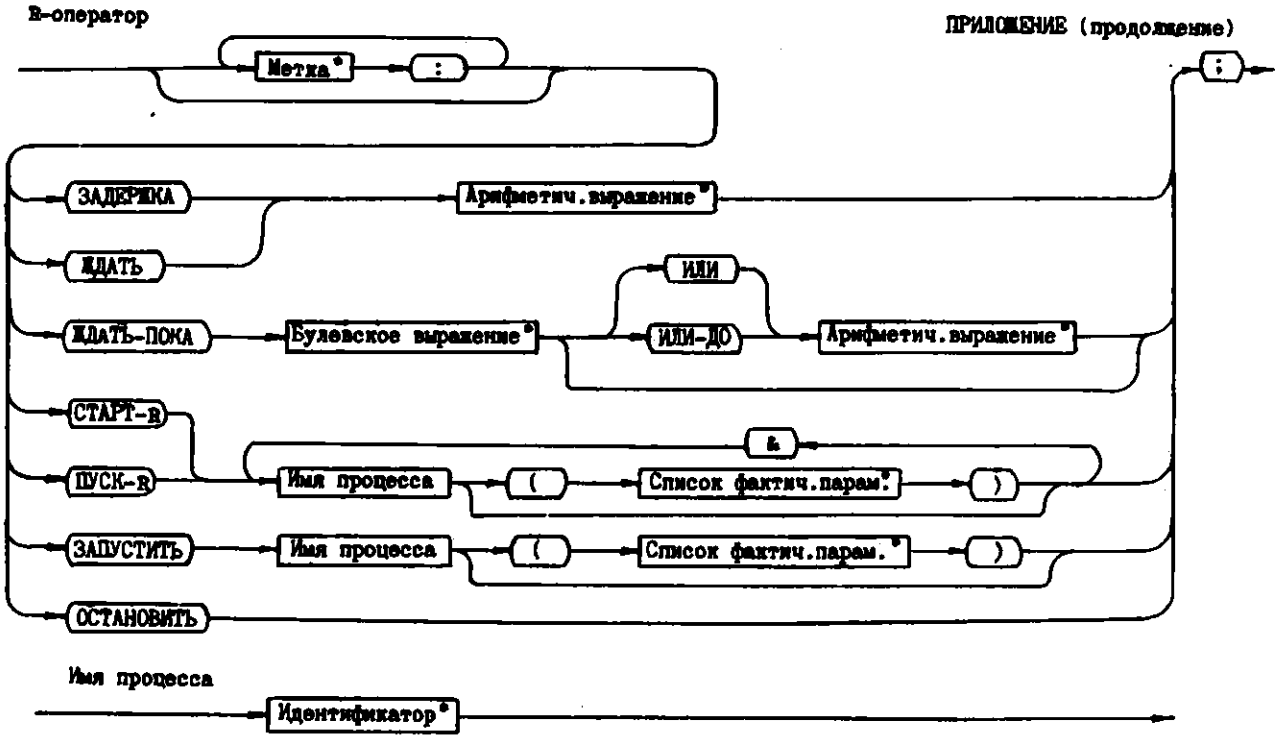
P-процедура



P-оператор




L-оператор
Метка*
:
;
ОЧЕРЕДЬ
Имя процесса
,
СТАРТ
ПУСК
НАЧАТЬ
&
Имя процесса
(
Список фактич.парам.*
)
ЗАПУСТИТЬ
Имя процесса
(
Список фактич.парам.*
)
В
ЧЕРЕЗ
ПРИ
Арифметич. выражение*
Булевское выражение*
ОСТАНОВИТЬ
Имя процесса
ЗАВЕРШИТЬ
ИСКЛЮЧИТЬ
ВКЛЮЧИТЬ
(
Имя процесса
:
,
Имя процесса
)
&
СТОП


R-оператор
Метка*
:
ЗАДЕРЖКА
Арифметич.выражение*
ЖДАТЬ
ЖДАТЬ-ПОКА
Булевское выражение*
ИЛИ
ИЛИ-ДО
Арифметич.выражение*
СТАРТ-R
ПУСК-R
&
Имя процесса
(
Список фактич.парам.*
)
ЗАПУСТИТЬ
Имя процесса
(
Список фактич.парам.*
)
ОСТАНОВИТЬ
ПРИЛОЖЕНИЕ (продолжение)
;
Имя процесса
Идентификатор*