
Технологическое программирование в ПТК «САРГОН». **Описание алгоритмов переключения**

Выпуск 3'2001

В.А. Менделевич

<i>Описание</i>	<i>непроцедурного</i>	<i>языка</i>
<i>технологического</i>	<i>программирования</i>	<i>(НАВТ).</i>

С конца XVIII века для упрощения описания сложных систем в науке и технике стали особо выделять стационарные состояния систем. При этом описание решения задачи разбивается на две части: статическое для стационарных состояний и динамическое для переходов из одних стационарных состояний в другие. Традиционные языки программирования (в том числе стандартные ИЕС 1131-3) используют динамические средства описания, как для динамических процессов, так и для стационарных состояний, что заставляет разработчика программы осуществлять «ручную компиляцию» статического представления в динамическое.

Принципиальным отличием языка технологического программирования комплекса «САРГОН» от языков предыдущих поколений состоит в наличии явных, интуитивно понятных средств как статического, так и динамического описаний. Причем их взаимная стыковка перекладывается с разработчика технологической программы на систему программирования.

Автоматные модели в последнее время находят все большее применение как основа создания языков программирования, т.к. теория автоматов предоставляет удобные средства описания стационарных состояний и переходных процессов. Однако, формы описания переходных процессов, принятая в теории автоматов, не понятна технологам без специального «алгоритмического» образования.

В комплексе «САРГОН» понятность программы для технолога была достигнута путем упрощения стандартных форм описания и автоматизации их взаимосогласования. При этом была сохранена математическая строгость автоматного представления, что обеспечивает высокую надежность программ.

Таблицы, описывающие статические зависимости, были подробно рассмотрены в первой статье. В данной статье рассматриваются средства динамического описания решения задач АСУТП - диаграммы переключений.

Диаграмма переключений

Важной частью языка «НАВТ» является описание алгоритмов многошаговых переключений типовых элементов (типовых моделей) из одного установившегося состояния в другое. Наиболее естественной для человека формой подобных описаний является диаграмма или блок-схема, поэтому в «НАВТ» была использована такая форма: «**диаграмма переключений**».

Структура диаграммы и набор ее элементов разработаны как объединение лучших черт языка программирования SFC (из ИЕС1131-3) и классического стандарта на блок-схемы. Попытка использования SFC в чистом виде не удалась из-за недостаточной технологической наглядности конструкций языка.

В предыдущей статье были описаны основные понятия, связанные с диаграммами

переключений.

Для удобства написания и чтения алгоритмов переключения используются шаги нескольких типов. При разработке типового набора шагов использовался многолетний опыт работы наших специалистов по созданию систем автоматизации и основополагающий принцип «САРГОН»: **максимально просто должны описываться наиболее часто выполняемые операции.**

В результате разделения описаний нормального хода процесса и обработки исключительных ситуаций в языке «НАВТ» (объяснено в первой статье), большинство диаграмм переключения представляет собой простую цепочку шагов и очевидны при составлении и чтении.

Для эффективного описания переключения на диаграмме должны быть отображены:

- условия начала переключения;
- результат переключения (состояния в которых может оказаться элемент в результате выполнения переключения);
- последовательность шагов переключения;
- содержание каждого шага.

Диаграммы переключения состоят из шагов переключения и соединений между ними.

Соединения типа «безусловный переход», «или», «и» имеют стандартный смысл и не требуют особых пояснений.

Шаг переключения

Шаг переключения содержит описание набора действий и условий их выполнения. Сложности внутренней структуры шага переключения «НАВТ» скрыты от разработчика, что обеспечивает простоту описания сложных переключений. Логические и арифметические выражения, записанные в шаге, оперируют только параметрами типового элемента (входными, выходными, коэффициентами) и его исполнителями.

Структура технологического шага

Технологический шаг состоит из блоков трех типов:

- «вход в шаг» - блок действий, выполняемых однократно при входе в шаг (активизации шага);
- «внутренний» - блок действий, выполняемых циклически до завершения шага (в течение всего времени активности шага);
- «выход из шага» - блок действий, выполняемых при выходе из шага.

Каждый шаг может иметь несколько условий выхода, соответствующих альтернативным ветвям алгоритма. Каждому выходу соответствует свой блок типа 3. Внутришаговый блок может содержать: вычисляемое выражение, имя исполнителя и код команды, условие перехода к новому шагу.

Для написания выражений используется язык ST стандарта IEC1131.

Типизация технологических шагов

При составлении перечня типов технологических шагов учитывались два основных требования:

- набор шагов должен позволять описывать любой алгоритм переключения;
- технологический смысл шага должен быть максимально очевиден.

Предусмотрены следующие типы технологических шагов:

1. **Начальный** - указывает установившееся состояние модели в начале

переключения (исходное). Условием выхода из шага (т.е. началом переключения) является получение команды, указанной в выходном блоке шага.

2. **Конечный** - указывает установившееся состояние модели после завершения переключения (конечное). В разветвляющемся переключении может быть несколько конечных шагов, каждый из которых соответствует одному из возможных конечных состояний модели.

3. **Безусловный** - содержит набор однократно выполняемых операторов и выдаваемых команд, описанных во входном блоке шага. После однократного выполнения шага осуществляется безусловный переход к следующему шагу переключения.

4. **Ожидание выполнения условия** - шаг, у которого присутствует одно условие выхода. Могут быть также заданы действия, выполняемые во время активности шага и при выходе из шага.

5. **Задержка времени (частный случай предыдущего)** - шаг ожидания условия, у которого условием выхода является истечение установленного времени задержки.

6. **Исполнение команды** - шаг, у которого во входном блоке задана команда исполнителю, а в единственном выходном блоке - условие выхода, контролирующее окончание выполнения команды этим исполнителем.

7. **Условный** - однократно производимый выбор дальнейшего пути выполнения переключения. Во входном блоке содержит группу операторов, вычисляющих критерий выбора ветви переключения. Число значений критерия выбора равно числу ветвей и, соответственно, числу выходных блоков (как правило, ветвей 2 - «да» и «нет»). В каждом выходном блоке указано значение критерия выбора соответствующей ветви.

8. **Макрошаг** - условное обозначение технологической операции, включающей в себя несколько шагов.

Рассмотрим подробнее наиболее характерный шаг «НАВТ» - шаг исполнения команды. Более половины шагов типовой программы на языке «НАВТ» имеют именно этот тип, который наглядно объединяет в себе выдаваемую команду и проверку условия ее выполнения. Например, шаг, описывающий закрытие клапана в алгоритме функционально-группового управления (ФГУ), имеет следующий вид:



При выполнении данного шага программному объекту, управляющему работой клапана П1, будет послана команда на закрытие, после чего программа переключения будет ожидать когда клапан сообщит, что он закрыт. После этого шаг будет считаться завершенным и программа продолжит выполнение переключения.

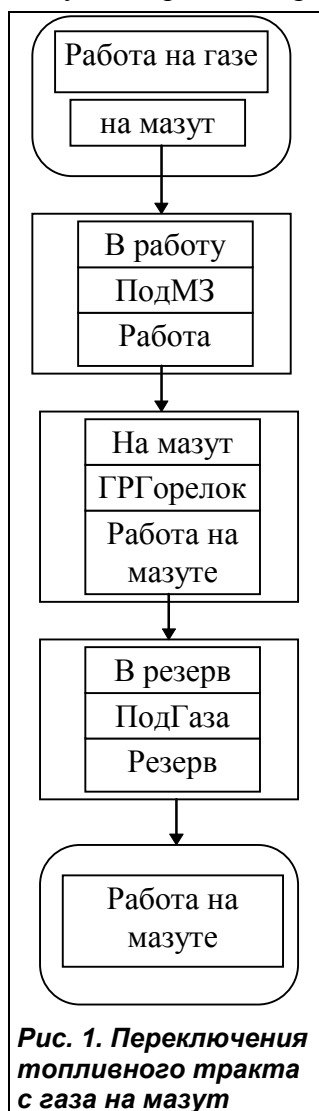
Объектная структура языка позволяет не рассматривать в алгоритме ФГУ, каким образом модель клапана определяет, что он выполнил команду - это описывается в типовой модели клапана (например, он может закрываться с уплотнением или без него: алгоритмы управления клапанами будут разными, а на работу алгоритма ФГУ это не повлияет).

Гигантские диаграммы, свойственные описаниям ФГУ на языках стандарта IEC-1131 (даже алгоритм розжига водогрейного котла занимает на SFC плотно заполненный лист формата А1), при программировании на «НАВТ» существенно упрощаются.

В программном комплексе «САРГОН» все объекты от клапана до энергоблока обрабатываются однотипно, что позволяет на каждом уровне управления (исполнительное устройство, функциональный узел, функциональная группа, технологическая установка, энергоблок) написать короткий и технологически прозрачный алгоритм даже для

переключения энергоблока в целом.

Например, алгоритм переключения топливного тракта газомазутного котла с газа на мазут изображен на рис.7



«ПодачаПГ» (подача природного газа), и «ПодачаМаз» (подача мазута) являются функциональными узлами, поэтому выполнение команды каждым из них приведет к последовательности переключений исполнительных устройств, но структура программы на всех уровнях остается такой же простой и технологически очевидной, как для клапана и топливного тракта котла.

Из приведенных примеров видна также высокая степень независимости получаемых программ от конкретных объектов управления – написанные в конкретном проекте программные элементы имеют выраженные «библиотечные» свойства даже без специальных усилий разработчика.

Параметры переключения

Для удобства и однотипности описания автоматических многошаговых переключений алгоритмическая модель в «САРГОН» содержит стандартный набор параметров, характеризующих переключение, значение которых автоматически отслеживается исполняющей системой. Это позволяет разработчику сосредоточить внимание на технологическом смысле переключения.

Типовыми параметрами переключения являются:

- последнее установившееся состояние элемента;
- текущий «ШагПереключения»;
- «ВремяШага».

Цель переключения определяется исполняемой моделью командой.

При достижении нового установившегося состояния «ШагПереключения» и «ВремяШага» обнуляются, а установившееся состояние фиксируется. Фиксация установившегося состояния производится как при выполнении автоматического алгоритма переключения, так и при дистанционном управлении отдельными исполнительными устройствами. «ВремяШага» обнуляется также в момент начала каждого шага. Сброс команды производится моделью после ее выполнения, при обнаружении невозможности ее выполнения или при получении более приоритетной команды.

Описанный способ автоматического вычисления переменных переключения позволяет в любой момент времени точно определить не только установившееся, но и переходное состояние элемента.

Средства разработки программ

Простота и наглядность типовых форм описания программ на языке «НАВТ» позволяет организовать для разработчика прикладных программ интерфейс, обеспечивающий эффективное написание программ и снижающий количество ошибок.

Структура диаграммы «рисует» специализированным графическим редактором, который позволяет размещать на экране типовые шаги переключения, устанавливать связи между ними и вызывает формы заполнения шагов.

Формы заполнения индивидуальны для каждого типа шага, что минимизирует число

операций, выполняемых разработчиком при заполнении шага.

Например, для наиболее часто используемого шага **исполнения команды** заполнение сводится к трехкратному выбору значения из списка:

- имени исполнителя из списка исполнителей данной типовой модели;
- названия команды исполнителю из списка возможных команд типовой модели выбранного исполнителя;
- названия конечного состояния исполнителя из списка возможных состояний типовой модели выбранного исполнителя.

Все списки включают небольшое число элементов (как правило, не более 10).