УДК 681.3.06 : 62-507

А.А. Шалыто, д-р техн. наук, **Н.И.Туккель**

Проектирование программного обеспечения системы управления дизель-генераторами на основе автоматного подхода

В работе [1] были изложены основные положения технологии автоматного программирования.

В настоящей работе применение этой технологии демонстрируется на примере проектирования программного обеспечения системы управления двумя дизель-генераторами, функционирующими по одинаковым алгоритмам. Система управления в целом содержит около 50 дискретных входов, 50 аналоговых входов, 50 дискретных выходов, до 20 одновременно активных выдержек времени и 5 видеокадров.

Программы выполнены для операционной системы QNX 4.25 и графической оболочки Photon 1.14.

Наряду с программным обеспечением системы управления с помощью той же технологии создана программная модель объекта управления. Так как и система управления, и модель объекта реализованы на одной и той же персональной ЭВМ, то эту реализацию можно рассматривать как понятийный тренажер для обучения операторов.

Основные исходные данные при проектировании системы содержатся в техническом задании, включающем как словесное описание, так и некоторые таблицы. В силу того, что при проектировании использовалась и другая информация, а сложное программное обеспечение словами (вербально) не описать, то техническое задание (или его фрагменты) применяется в документации только как комментарий к алгоритмам, представленным в виде графов переходов, и реализующим их программам.

В настоящей работе используется не объектное [2], а процедурное программирование. Совместное применение процедурного и автоматного подходов в работе [3] названо "программирование с явным выделением состояний".

В рамках предлагаемой технологии автоматы используются при спецификации, программировании и для протоколирования. Поэтому на основании обзора, выполненного в работе [4], можно утверждать, что предлагаемый подход является единственным, в котором на базе одной модели осуществляется решение трех основных задач, возникающих при разработке программного обеспечения: проектирования, реализации и отладки.

При этом отметим, что в общем случае в рамках предлагаемого подхода программа состоит из двух частей: системонезависимой (например, от операционной системы) и системозависимой.

Разработка системонезависимой части

В предлагаемом подходе системонезависимую часть образуют автоматы. В начале проектирования по словесному описанию эвристически строится схема их взаимодействия (рис. 1). При этом, в первую очередь, в иерархии выделяются головные автоматы. Некоторые из них в дальнейшем детализируются за счет вложенных автоматов. Для рассматриваемого в данной работе примера построение схемы взаимодействия завершилось выделением 31 автомата. Эти автоматы взаимодействуют тремя способами: по вложенности, через обмен номерами состояний и по вызываемости.

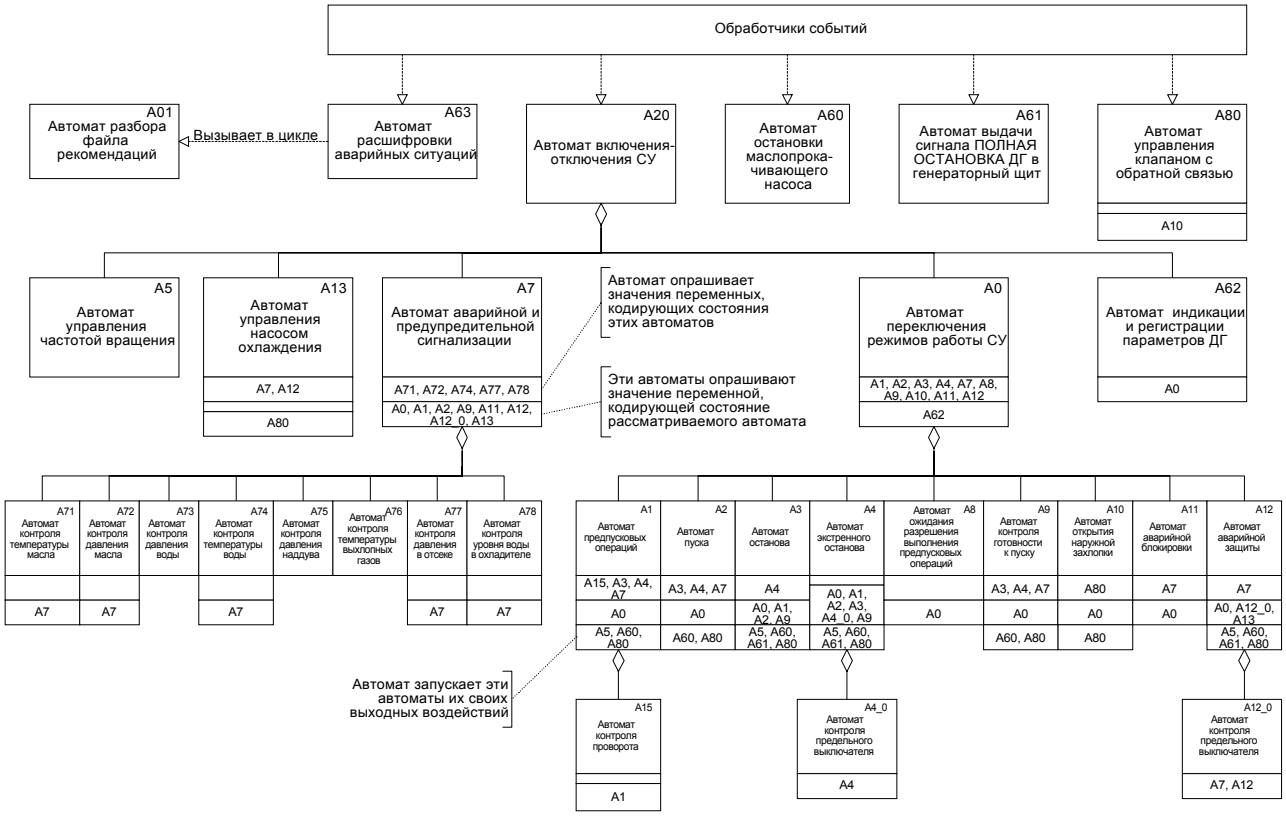

Рис. 1. Схема взаимодействия автоматов

Ниже в качестве примера рассматриваются два взаимосвязанных автомата, первый из которых управляет режимами работы системы, а второй обеспечивает управление в режиме предпусковых операций.

Для каждого автомата разрабатываются четыре документа:

- словесное описание (комментарий);
- схема связей автомата;
- граф переходов автомата;
- текст функции, реализующей автомат.

Схема связей автомата специфицирует его интерфейс. На этой схеме указываются источники и приемники информации и для каждого входного и выходного воздействия приводятся его символическое обозначение и полное название на русском языке. На схеме перечисляются все события, входные переменные и выходные воздействия. Если это возможно, схема располагается на одном листе формата А4 вместе с графом переходов. В отдельных случаях схема может быть заменена табличным перечнем входных и выходных воздействий автомата, также располагаемым на одном листе с графом переходов, напоминая спецификацию на конструкторских чертежах.

Применение схемы связей автомата сильно отличает предлагаемую технологию от других тем, что не предпринимается попытка сделать графы переходов или тексты программ самодокументируемыми. По мнению авторов, использовать смысловые идентификаторы входных и выходных воздействий на графах переходов при сложной логике невозможно, а комментарии в текстах программ при сложной логике также мало что объясняют. При этом единственными смысловыми обозначениями в графе переходов являются названия состояний автомата.

Нотация, используемая при построении графа переходов, приведена на рис. 2.

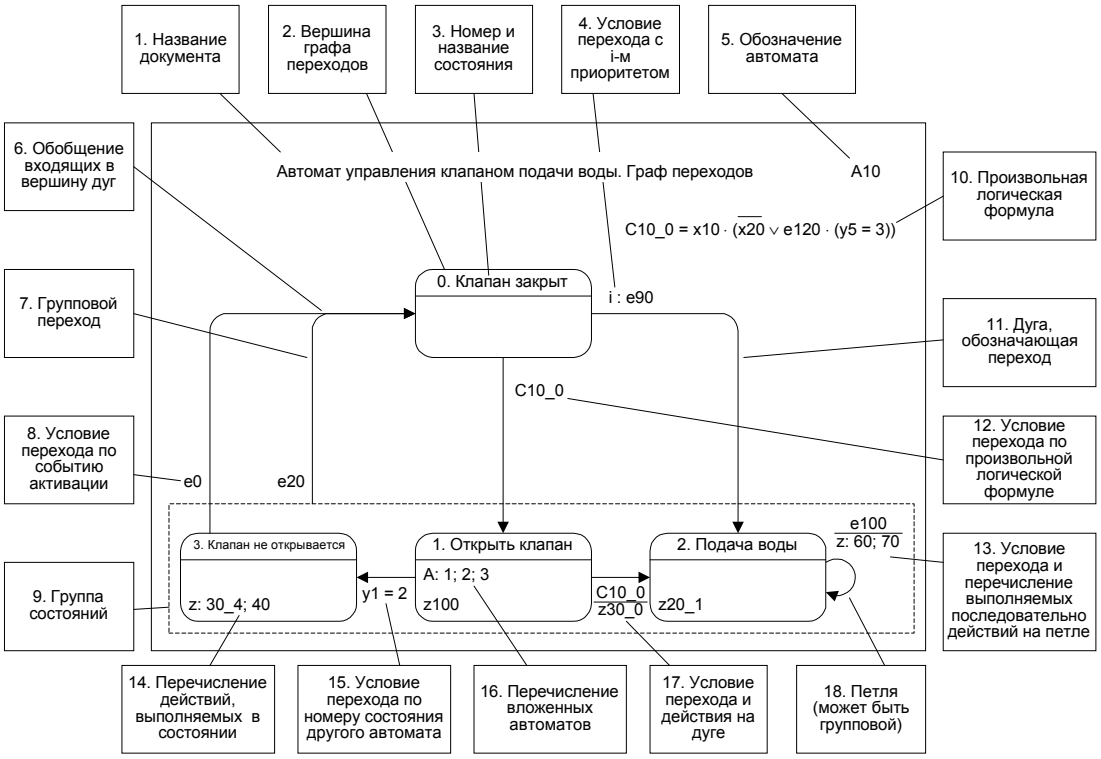

Рис. 2. Нотация графа переходов

В силу простоты и формализованности этой нотации каждый граф переходов может быть вручную или автоматически реализован по шаблону [3].

При ручном способе реализации возможны механические ошибки. Однако на практике в процессе ручной реализации, которая занимает не очень много времени, часто обнаруживаются смысловые ошибки.

При автоматической реализации механические ошибки исключены, но исчезает дополнительный шанс обнаружить и исправить смысловые ошибки. Одно из преимуществ автоматической реализации заключается в том, что названия состояний могут быть перенесены в текст программы в качестве комментариев и для их протоколирования, что весьма трудоемко при ручной реализации.

В настоящей работе выполнялась ручная реализация автоматов. Отметим, что в настоящее время существует конвертор, генерирующий по графу переходов (построенному в графическом редакторе "Visio") текст функции на языке Си [5]. При этом граф переходов строится в соответствии с приведенной нотацией, а текст реализующей его функции соответствует применяемому шаблону.

Перейдем к рассмотрению примера. Для сокращения объема статьи текст программы приведен только для реализации второго автомата.

Автомат переключения режимов работы системы управления

Автомат переключения режимов работы реализует основной алгоритм управления. Автоматы, вложенные в его состояния, обеспечивают детализацию алгоритма в соответствующих режимах. Перечислим эти режимы, каждому из которых соответствует выделенное состояние автомата.

1. Дизель остановлен. В рассматриваемом режиме осуществляется проверка условий начала выполнения алгоритма предпусковых операций. Если при нахождении системы управления в этом режиме дизель-генератор был запущен в обход нее, то система отреагирует на это переходом в установившийся режим. Срабатывание одного из алгоритмов аварийной и предупредительной сигнализации в этом режиме приводит к переходу системы в режим аварийной блокировки.

2. Предпусковые операции. В рассматриваемом режиме выполняется алгоритм предпусковых операций.

3. К пуску готов. Режим соответствует удачному завершению выполнения алгоритма предпусковых операций. В этом режиме система проверяет сохранились ли условия, разрешающие пуск. При нажатии оператором кнопки "Пуск" начинается выполнение алгоритма пуска.

4. Выполняется пуск. В рассматриваемом режиме выполняется алгоритм пуска, при успешном осуществлении которого система переходит в установившийся режим.

5. Установившийся режим. В этом режиме при необходимости выполняется алгоритм открытия наружной захлопки, реализуемый автоматом А10. Выход из этого режима происходит в начале выполнения одного из видов останова, при срабатывании алгоритма аварийной защиты или при останове дизель-генератора в обход системы управления.

6. Останов. В рассматриваемом режиме выполняется алгоритм останова. Указанный алгоритм может быть прерван в случае экстренного останова.

7. Экстренный останов. В рассматриваемом режиме выполняется алгоритм экстренного останова. После завершения этого алгоритма система переходит в режим аварийной блокировки.

8. Аварийная блокировка. В этом режиме система ждет пока оператор подтвердит свою реакцию на аварию нажатием кнопки "Квитирование" и, устранив аварию, нажмет кнопку "Разблокировка".

9. Аварийная защита. Этот режим аналогичен режиму экстренного останова.

Схема связей автомата приведена на рис. 3.

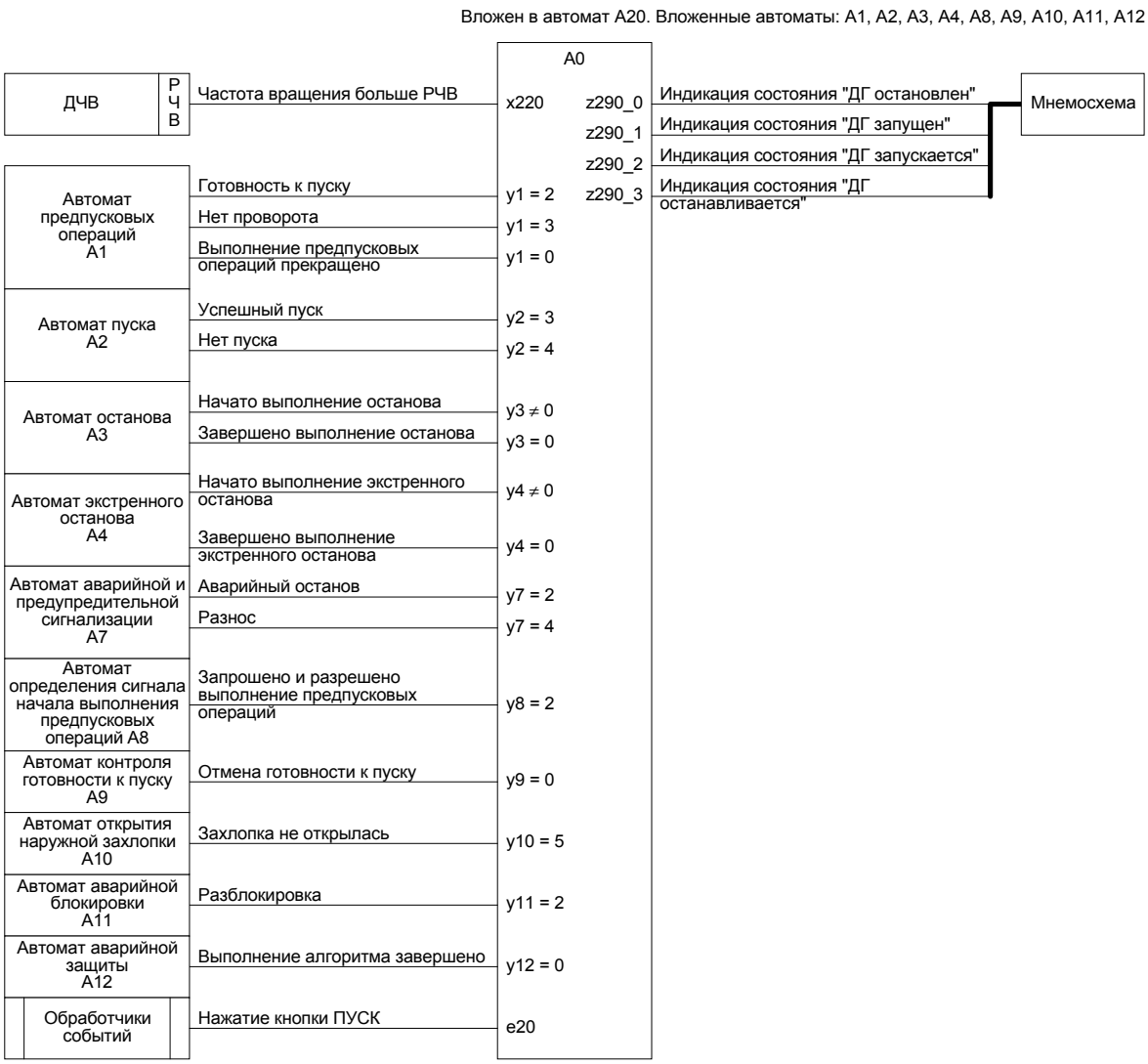

Рис. 3. Схема связей автомата А0

Граф переходов автомата приведен на рис. 4.

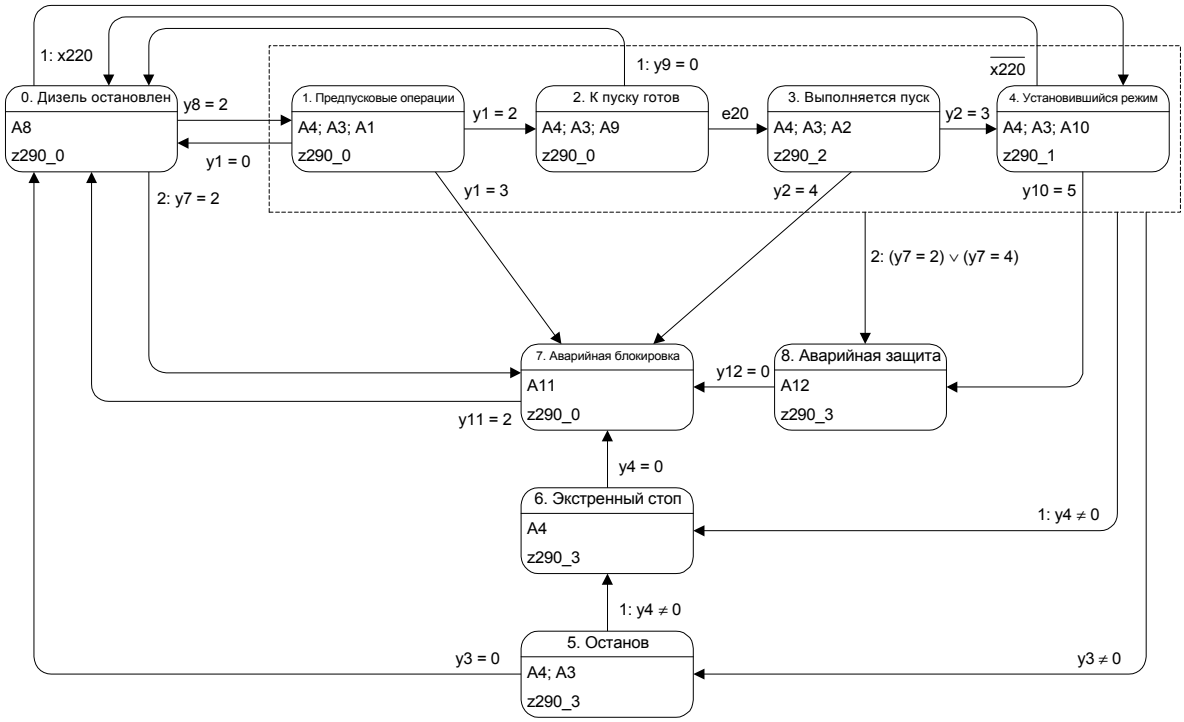

Рис. 4. Граф переходов автомата A0

Автомат предпусковых операций

Приведем фрагмент технического задания, который был первоначально использован при построении рассматриваемого автомата.

1. Перечислим сигналы, блокирующие проведение предпусковых операций.

1.1. Обобщенный сигнал "Блокировка пуска" из СУ ОКС. При наличии этого сигнала выдается сигнал "Подготовка пуска" в указанную систему.

1.2. Сигнал "Механизм валопроворотный не отключен".

1.3. Сигнал "Предельный выключатель".

1.4. Дизель не остановлен — частота вращения превышает рабочую частоту.

1.5. Команда на выполнение любого вида останова.

1.6. Наличие сигнала "Вода в охладителе" (аварийный уровень).

2. При наличии хотя бы одного из сигналов, указанных в пп. 1.2...1.6, система управления не должна принимать управляющую команду подготовки к пуску.

3. При нажатии оператором кнопки "Подготовка к пуску" выдается сигнал в СУ ОКС. После снятия сигнала "Блокировка пуска" из СУ ОКС и при отсутствии блокирующих сигналов по пп. 1.2...1.6, система должна выполнить перечисленные ниже действия.

3.1. Запомнить команду на подготовку к пуску.

3.2. Включить табло "Подготовка к пуску".

3.3. Выдать команду на включение маслопрокачивающего насоса, замкнув контакт в цепи его магнитного пускателя.

3.4. Включить табло "Прокачка маслом" при замыкании контакта пускателя маслопрокачивающего насоса.

3.5. Подать питание на электромагнит стопа регулятора частоты вращения.

3.6. Подать команду на открытие клапана подачи воздуха к дизель-генератору.

3.7. Подать команду на открытие клапана воздуха низкого давления.

3.8. Включить табло "Проворот".

3.9. Включить контроль времени проворота (60 с).

3.10. Выдать команду в регулятор частоты вращения на установку первой ступени частоты и через 4 с снять эту команду.

4. После получения 3-х импульсов от датчика проворота коленчатого вала (до истечения времени по п. 3.9) система управления должна выполнить перечисленные ниже действия.

4.1. Снять питание с электромагнита клапана воздуха низкого давления.

4.2. Отключить табло "Проворот".

4.3. Отключить контроль времени проворота (60 с).

5. При превышении давлением масла предпускового давления и отработки команды на уменьшение заданной частоты вращения до первой ступени, система управления должна выполнить перечисленные ниже действия.

5.1. Снять команду на включение маслопрокачивающего насоса.

5.2. Снять команды на подготовку к пуску, перечисленные в п. 3.

5.3. Отключить табло "Подготовка к пуску".

5.4. Включить табло "Пуск разрешен".

5.5. Включить память завершения предпусковых операций и дать разрешение на выполнение пуска.

6. В случае снижения давления масла ниже предпускового давления или увеличения затяжки пружины регулятора частоты вращения до третьей позиции или появления блокирующих сигналов по п. 1.1–1.3, 1.5, 1.6 система управления должна выполнить перечисленные ниже действия.

6.1. Отключить память завершения предпусковых операций.

6.2. Отключить табло "Пуск разрешен".

6.3. Разомкнуть контакт остановки маслопрокачивающего насоса и через 4 с вновь замкнуть его.

6.4. Снять питание с электромагнита остановки регулятора частоты вращения.

6.5. Отключить табло "Прокачка маслом" при размыкании контакта пускателя маслопрокачивающего насоса.

6.6. Снять команду на открытие клапана подачи воздуха к дизель-генератору.

7. Если по истечении времени (п. 3.9) от датчика проворота коленвала не поступило 3-х импульсов, то система управления должна выполнить перечисленные ниже действия.

7.1. Включить аварийное табло "Нет проворота".

7.2. Включить обобщенный сигнал звуковой сигнализации.

7.3. Отключить табло "Подготовка к пуску".

7.4. Снять питание с электромагнита клапана пускового воздуха низкого давления.

7.5. Отключить контроль времени проворота (60 с).

7.6. Снять команду на включение маслопрокачивающего насоса.

7.7. Разомкнуть контакт остановки маслопрокачивающего насоса и через 4 с вновь замкнуть этот контакт.

7.8. Отключить табло "Прокачка маслом" при размыкании контакта пускателя маслопрокачивающего насоса.

7.9. Отключить память команды на подготовку к пуску.

7.10. Снять команду на открытие клапана подачи воздуха к дизель-генератору.

8. Отключение обобщенной звуковой сигнализации и перевод аварийного табло "Нет проворота" в постоянное свечение производится нажатием кнопки "Квитирование".

9. Разблокировка системы управления и отключение табло "Нет проворота" производится нажатием кнопки "Разблокировка".

Весьма насыщенная схема связей автомата приведена на рис. 5. Обратим внимание, что она напоминает общую схему, которая всегда разрабатывается при конструировании аппаратуры.

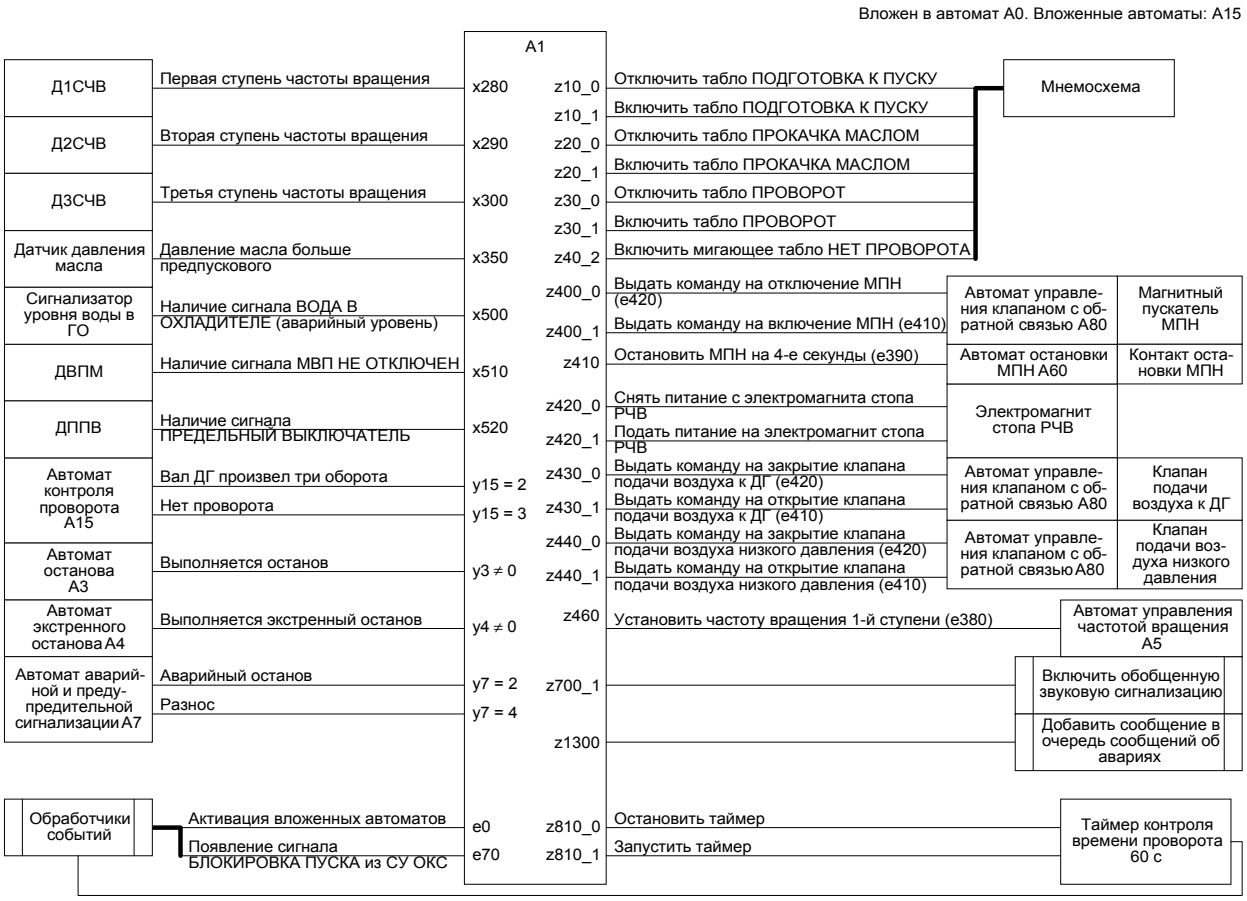

Рис. 5. Схема связей автомата A1

Отметим, что в графе переходов этого автомата (рис. 6) пометки некоторых дуг и вершин настолько сложны, что не могут быть в обозримом виде записаны при использовании смысловых обозначений для входных и выходных воздействий, как это предлагается в большинстве других существующих на сегодняшний день подходов к проектированию программ (например, при использовании языка UML [6]). Применение же кратких символьных обозначений, расшифровка которых приведена на схеме связей (рис. 5), позволяет компактно записывать сложные условия переходов и длинные перечисления формируемых выходных воздействий.

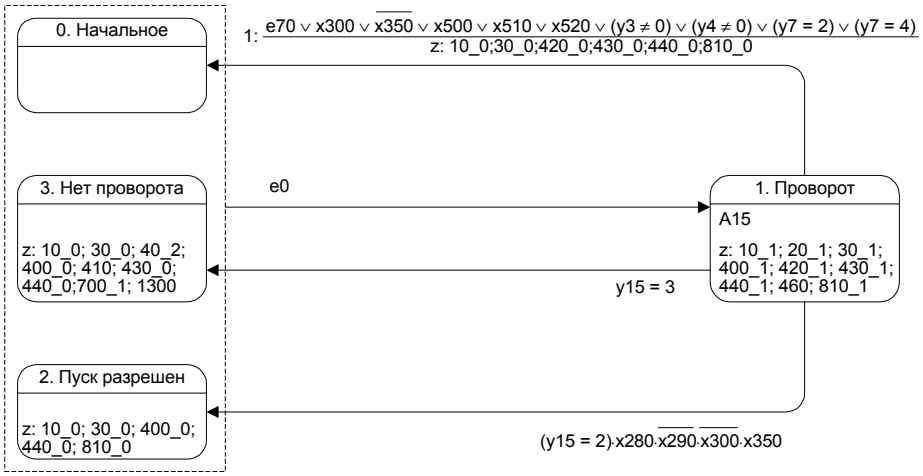

Рис. 6. Граф переходов автомата A1

Текст функции, реализующей этот автомат, приведен в листинге 1.

ЛИСТИНГ 1. Функция автомата A1

```
void A1( int e, dg_t *dg )
{
  int y_old = dg->y1 ;
```

```
  #ifdef GRAPH_EVENTS_LOGGING
    log_exec( dg, "A1", y_old, e ) ;
  #endif

  switch( dg->y1 )
  {
    case 0:
      if( e == 0 )                                  dg->y1 = 1 ;
    break ;
    case 1:
      A15( e, dg ) ;
      if(    e == 70 || x300(dg) || !x350(dg)
          || x500(dg) || x510(dg) || x520(dg)
          || dg->y3 != 0 || dg->y4 != 0
          || dg->y7 == 2 || dg->y7 == 4 )
      { z10_0(dg) ; z30_0(dg) ; z420_0(dg) ;
        z430_0(dg) ; z440_0(dg) ; z810_0(dg) ;
                                                    dg->y1 = 0 ; }
      else
      if( dg->y15 == 3 )
                                                    dg->y1 = 3 ;
      else
      if(    dg->y15 == 2 && x350(dg)
          && x280(dg) && !x290(dg) && !x300(dg) )
                                                    dg->y1 = 2 ;
    break ;
    case 2:
      if( e == 0 )                                  dg->y1 = 1 ;
    break ;
    case 3:
      if( e == 0 )                                  dg->y1 = 1 ;
    break ;
    default:
      #ifdef GRAPH_ERRORS_LOGGING
        log_write( LOG_GRAPH_ERROR, dg->number,
                   "ERROR IN A1: unknown state number!", 0 ) ;
      #endif
    break ;
  } ;

  if( y_old == dg->y1 ) goto A1_end ;

  #ifdef GRAPH_TRANS_LOGGING
    log_trans( dg, "A1", y_old, dg->y1 ) ;
  #endif
  #ifdef DEBUG_FRAME
    update_debug() ;
  #endif

  switch( dg->y1 )
  {
    case 1:
      A15( 0, dg ) ;
      z10_1(dg) ; z20_1(dg) ; z30_1(dg) ; z400_1(dg) ; z420_1(dg) ;
      z430_1(dg) ; z440_1(dg) ; z460(dg) ; z810_1(dg) ;
    break ;
    case 2:
      z10_0(dg) ; z30_0(dg) ; z400_0(dg) ; z440_0(dg) ; z810_0(dg) ;
    break ;
    case 3:
      z10_0(dg) ; z30_0(dg) ; z40_2(dg) ; z400_0(dg) ; z430_0(dg) ;
      z440_0(dg) ; z410(dg) ;
      z700_1(dg) ; z1300(dg, MSG_NO_TURN) ;
    break ;
  } ;
  A1_end: ;
  #ifdef GRAPH_ENDS_LOGGING
    log_end( dg, "A1", dg->y1, e ) ;
  #endif
} ;
```

Разработка системозависимой части

Системозависимая часть состоит из обработчиков событий, функций, реализующих входные переменные и выходные воздействия и вспомогательных модулей (в том числе модулей, реализующих интерфейс оператора).

В качестве примера приведем обработчик события e10, вызываемого при нажатии кнопки "Подготовка к пуску" (листинг 2).

ЛИСТИНГ 2. Пример обработчика события

```
int e10( PtWidget_t *widget, ApInfo_t *apinfo, PtCallbackInfo_t *cbinfo )
{
  if( widget == ABW_dg1_prepare_btn )
    A20( 10, &dgs[DG1] ) ;
  else if( widget == ABW_dg2_prepare_btn )
    A20( 10, &dgs[DG2] ) ;
  return Pt_CONTINUE ;
} ;
```

В качестве следующего примера приведем реализацию входной переменной x20, которая опрашивает положение переключателя "Защита отключена" (листинг 3).

ЛИСТИНГ 3. Пример входной переменной

```
int x20( dg_t *dg )
{
  int result = 0 ;
  result = dg->number == 2 ?
             is_wgt_set(ABW_dg2_protection_btn) :
             is_wgt_set(ABW_dg1_protection_btn) ;
  #ifdef INPUTS_LOGGING
    log_input( dg, "x20 - защита включена -", result ) ;
  #endif
  return result ;
} ;
```

Следующий пример связан с реализацией выходного воздействия z10, выключающего (z10_0) и включающего (z10_1) табло "Подготовка к пуску" (листинг 4).

ЛИСТИНГ 4. Пример выходной переменной

```
void z10_0( dg_t *dg )
{
  #ifdef ACTIONS_LOGGING
    log_write( LOG_ACTION, dg->number,
               "z10_0. Отключить табло ПОДГОТОВКА К ПУСКУ.", 0 ) ;
  #endif
  tab_off( dg->prepare_tab ) ;
} ;

void z10_1( dg_t *dg )
{
  #ifdef ACTIONS_LOGGING
    log_write( LOG_ACTION, dg->number,
               "z10_1. Включить табло ПОДГОТОВКА К ПУСКУ.", 0 ) ;
  #endif
  tab_on( dg->prepare_tab, on_tab_color ) ;
} ;
```

К вспомогательным модулям относится, например, модуль управления таймерами. В зависимости от характера алгоритма, реализуемого вспомогательными модулями, они также могут содержать автоматы.

Отладка и тестирование

При использовании предлагаемой технологии возможны три способа отладки. Первый способ является традиционным и состоит в применении отладчика. Второй способ состоит в наблюдении за значениями переменных состояний каждого из автоматов (свойство

наблюдаемости автоматов). Третий способ основан на автоматическом протоколировании реакции системы на входные воздействия.

При этом если первый способ целесообразно использовать для системозависимой части, то второй и третий — для системонезависимой.

Протоколирование можно выполнять с любой степенью подробности. При этом можно выделить две разновидности протоколов: "полные" и "короткие". В "полных" протоколах указываются события, запуск автоматов, их состояния в момент запуска, переходы в новые состояния, завершение работы автоматов, значения входных переменных, выходные воздействия и время начала выполнения каждого из них. "Короткие" протоколы содержат только события и инициируемые ими выходные воздействия, интересующие заказчика.

В листинге 5 приведен фрагмент полного протокола работы системы управления при обработке нажатия кнопки "Подготовка к пуску" (событие e10).

ЛИСТИНГ 5. Фрагмент полного протокола обработки нажатия кнопки "Подготовка к пуску"

```
11:34:02.507{ DG1: A20: в состоянии 2 запущен с событием e10
11:34:02.507{ DG1: A7: в состоянии 0 запущен с событием e10
11:34:02.507{ DG1: A71: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x320 - температура масла меньше Тмм - вернул 0
11:34:02.507> DG1: x330 - температура масла больше Тмпр - вернул 0
11:34:02.507} DG1: A71: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A72: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x220 - частота вращения больше РЧВ - вернул 0
11:34:02.507} DG1: A72: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A73: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x220 - частота вращения больше РЧВ - вернул 0
11:34:02.507} DG1: A73: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A74: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x430 - температура воды меньше Твм - вернул 0
11:34:02.507> DG1: x440 - температура воды больше Твпр - вернул 0
11:34:02.507} DG1: A74: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A75: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x460 - давление наддува больше Рнпр - вернул 0
11:34:02.507} DG1: A75: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A76: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x470 - температура газов больше Твг - вернул 0
11:34:02.507} DG1: A76: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A77: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x480 - давление в отсеке ниже Ротс - вернул 0
11:34:02.507} DG1: A77: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A78: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x500 - сигнал ВОДА В ОХЛАДИТЕЛЕ - вернул 0
11:34:02.507} DG1: A78: завершил обработку события e10 в состоянии 0
11:34:02.507> DG1: x20 - защита включена - вернул 1
11:34:02.507> DG1: x260 - частота вращения больше ППРЕД - вернул 0
11:34:02.507} DG1: A7: завершил обработку события e10 в состоянии 0
11:34:02.507{ DG1: A0: в состоянии 0 запущен с событием e10
11:34:02.507{ DG1: A8: в состоянии 0 запущен с событием e10
11:34:02.507> DG1: x220 - частота вращения больше РЧВ - вернул 0
11:34:02.507> DG1: x500 - сигнал ВОДА В ОХЛАДИТЕЛЕ - вернул 0
11:34:02.507> DG1: x510 - сигнал МВП НЕ ОТКЛЮЧЕН - вернул 0
11:34:02.507> DG1: x520 - сигнал ПРЕДЕЛЬНЫЙ ВЫКЛЮЧАТЕЛЬ - вернул 0
11:34:02.507> DG1: x700 - сигнал БЛОКИРОВКА ПУСКА из СУ ОКС - вернул 0
11:34:02.507T DG1: A8: перешел из состояния 0 в состояние 2
11:34:02.507} DG1: A8: завершил обработку события e10 в состоянии 2
11:34:02.507T DG1: A0: перешел из состояния 0 в состояние 1
11:34:02.507{ DG1: A4: в состоянии 0 запущен с событием e0
11:34:02.507{ DG1: A4_0: в состоянии 0 запущен с событием e0
11:34:02.507} DG1: A4_0: завершил обработку события e0 в состоянии 0
11:34:02.507} DG1: A4: завершил обработку события e0 в состоянии 0
11:34:02.507{ DG1: A3: в состоянии 0 запущен с событием e0
11:34:02.507} DG1: A3: завершил обработку события e0 в состоянии 0
11:34:02.507{ DG1: A1: в состоянии 0 запущен с событием e0
11:34:02.507T DG1: A1: перешел из состояния 0 в состояние 1
11:34:02.507{ DG1: A15: в состоянии 0 запущен с событием e0
11:34:02.507* DG1: z800_0. Сбросить счетчик проворотов.
11:34:02.507} DG1: A15: завершил обработку события e0 в состоянии 0
11:34:02.507* DG1: z10_1. Включить табло ПОДГОТОВКА К ПУСКУ.
11:34:02.507* DG1: z20_1. Включить табло ПРОКАЧКА МАСЛОМ.
11:34:02.507* DG1: z30_1. Включить табло ПРОВОРОТ.
11:34:02.507* DG1: z400_1. Включить магнитный пускатель МПН.
11:34:02.507* DG1: z420_1. Подать питание на электромагнит стопа РЧВ.
11:34:02.507* DG1: z430_1. Открыть клапан подачи воздуха к ДГ.
11:34:02.507* DG1: z440_1. Открыть клапан подачи воздуха низкого давления.
```

```
11:34:02.507* DG1: z460. Выдать команду на установку частоты вращения первой ступени.
11:34:02.507{ DG1: A5: в состоянии 0 запущен с событием e380
11:34:02.507> DG1: x810 - заданная частота вращения выше установленной - вернул 1
11:34:02.507T DG1: A5: перешел из состояния 0 в состояние 2
11:34:02.507* DG1: z610_1. Увеличить установленную частоту РЧВ.
11:34:02.507} DG1: A5: завершил обработку события e380 в состоянии 2
11:34:02.507* DG1: z810_1. Запустить таймер контроля проворота.
11:34:02.507} DG1: A1: завершил обработку события e0 в состоянии 1
11:34:02.507* DG1: z290_0. Индикация состояния ДГ - остановлен.
11:34:02.507} DG1: A0: завершил обработку события e10 в состоянии 1
11:34:02.507{ DG1: A5: в состоянии 2 запущен с событием e10
11:34:02.507> DG1: x810 - заданная частота вращения выше установленной - вернул 1
11:34:02.517> DG1: x820 - заданная частота вращения ниже установленной - вернул 0
11:34:02.517} DG1: A5: завершил обработку события e10 в состоянии 2
11:34:02.517{ DG1: A13: в состоянии 0 запущен с событием e10
11:34:02.517} DG1: A13: завершил обработку события e10 в состоянии 0
11:34:02.517{ DG1: A62: в состоянии 0 запущен с событием e10
11:34:02.517T DG1: A62: перешел из состояния 0 в состояние 1
11:34:02.517* DG1: z1110_1. Запустить таймер регистрации параметров.
11:34:02.517* DG1: z1120_1. Запустить таймер индикации параметров.
11:34:02.517} DG1: A62: завершил обработку события e10 в состоянии 1
11:34:02.517} DG1: A20: завершил обработку события e10 в состоянии 2
```

Для доказательства соответствия программы техническому заданию при ее демонстрации заказчику и в прочих ситуациях, когда следует протоколировать только выходные воздействия, формируемые программой, можно использовать короткий протокол. Такой протокол для обработки нажатия кнопки "Подготовка к пуску" приведен в листинге 6.

ЛИСТИНГ 6. Фрагмент короткого протокола обработки нажатия кнопки "Подготовка к пуску"

```
11:41:06.188{ DG1: A20: в состоянии 2 запущен с событием e10
11:41:06.188* DG1: z10_1.  Включить табло ПОДГОТОВКА К ПУСКУ.
11:41:06.188* DG1: z20_1.  Включить табло ПРОКАЧКА МАСЛОМ.
11:41:06.188* DG1: z30_1.  Включить табло ПРОВОРОТ.
11:41:06.188* DG1: z400_1. Включить магнитный пускатель МПН.
11:41:06.188* DG1: z420_1. Подать питание на электромагнит стопа РЧВ.
11:41:06.188* DG1: z430_1. Открыть клапан подачи воздуха к ДГ.
11:41:06.188* DG1: z440_1. Открыть клапан подачи воздуха низкого давления.
11:41:06.188* DG1: z460.   Выдать команду на установку частоты вращения первой ступени.
11:41:06.188* DG1: z810_1. Запустить таймер контроля проворота.
11:41:06.188* DG1: z290_0. Индикация состояния ДГ - остановлен.
11:41:06.188} DG1: A20: завершил обработку события e10 в состоянии 2
```

Короткий протокол позволяет обнаружить несоответствие программы техническому заданию, а полный протокол — определить, какой автомат следует при этом откорректировать.

Обратим внимание, что практически все действия в рассмотренных протоколах выполняются за один квант времени. Поэтому протоколирование необходимо для того, чтобы понять, как работает программа при столь быстротекущих последовательностях действий, выполняемых в ней.

Заключение

Рассмотренный выше пример демонстрирует то, что разработанные программы реализуют весьма сложные алгоритмы управления. Их непросто понять, даже используя предложенную технологию, включающую различные схемы, диаграммы и протоколы. Авторам остается только догадываться, насколько возросла бы сложность программирования данной задачи и последующего понимания построенных программ при применении традиционного подхода, в котором для реализации логики используются флаги.

По-нашему мнению, для задач рассматриваемого класса использование объектно-ориентированного подхода не решило бы проблему понимания построенных программ.

Отметим также, что при столь сложной логике, как в рассмотренном примере, применение диаграмм взаимодействий и диаграмм состояний из UML практически невозможно.

Настоящая работа была выполнена в ФГУП "НПО "Аврора"" при поддержке Российского фонда фундаментальных исследований по гранту №02-07-90114 "Разработка технологии автоматного программирования".

Литература

1. Шалыто А.А., Туккель Н.И. Автоматный подход к созданию программного обеспечения для систем логического управления и "реактивных" систем // Системы управления и обработки информации. ФГУП "НПО "Аврора"". — 2000. — Вып. 2.
2. Туккель Н.И., Шалыто А.А. Система управления танком для игры Robocode. Объектно-ориентированное программирование с явным выделением состояний. Программная документация. **http://is.ifmo.ru**, раздел "Проекты".
3. Шалыто А.А., Туккель Н.И. Программирование с явным выделением состояний // Мир ПК. — 2001. — № 8, 9.
4. Шалыто А.А. Алгоритмизация и программирование для систем логического управления и реактивных систем // Автоматика и телемеханика. — 2001. — № 1.
5. Головешин А. Конвертор Visio — SWITCH. **http://is.ifmo.ru**, раздел "Последователи".
6. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. — М.: ДМК, 2000.