

УДК 681.3.06: 685.51

## ОБЪЕКТНО-СОБЫТИЙНЫЙ ПОДХОД К МОДЕЛИРОВАНИЮ АЛГОРИТМОВ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИХ МАШИН

Акунович С. И., Гончаров А. А.

Белорусский государственный технологический университет  
Белорусский национальный технический университет

Для всё усложняющегося технологического оборудования большой и большой вес приобретает применение IT-технологий при проектировании и реализации сложных управляющих устройств этим оборудованием. Предлагается способ проверки логических описаний таких устройств при проектировании и реализации процесса управления. Может быть использован при разработке и диагностировании управляющих программ.

### Введение

В настоящее время способность ученых, инженеров, других инженерно-технических работников поддерживать темпы научно-технического прогресса определяют информационные технологии. Знание этих технологий является также необходимым элементом подготовки специалистов высокой квалификации.

Это положение особенно существенно при проектировании и производстве сложных систем управления(или управляющих устройств), а также при обучении эффективному проектированию этих устройств.

Традиционно процесс создания сложных технических объектов, устройств состоит из нескольких этапов, важнейшим из которых является проектирование. Результатом этого этапа является проект некоторого изделия, устройства или технологической машины (ТМ).

С формальной точки зрения под проектом понимают некоторую совокупность взаимосвязанных документов, описывающих определенный объект или процесс [1].

В настоящее время ускорение разработки проектов достигается за счёт применения автоматизированных с помощью ЭВМ методов проектирования, которые также позволяют существенно повысить качество и уровень проектно-конструкторских работ. В таком случае проект **P**, составляющие его элементы и их совокупности, части проекта представляются терминами, понятиями операционной среды ЭВМ. Говорят, что проекты погружаются в среду ЭВМ (или вычислительной системы).

Определение 1. Некоторый проект **P**, погружённый в среду мультимедийной системы **SYS** и дополненный специальными объектами и органами управления, назовём моделью проекта **P** в **SYS** или гиперпроектом **GP**. Последний представляет собой отображение декартова произведения множеств **P** и **SYS**, то есть  $P \times SYS \rightarrow GP$ , где **P** и **SYS** определены выше;

$GP = \{PGP_1, PGP_2, \dots, PGP_{i1}, \dots, PGP_{i1}\}$  — множество объектов, называемых папками гиперпроекта;  $PGP_{i1} = \{SGP_{i1,1}, SGP_{i1,2}, \dots, SGP_{i1,i2}, \dots, SGP_{i1,i2}\}$  — *i*-я папка гиперпроекта, состоящая из множества объектов  $SGP_{i1,i2}$ , называемых его страницами, причём  $\{SGP_{i3}\}$  включается в **GP**, а **I3** — мощность этого множества; **I1** — мощность множества страниц *i1*-й папки;

$SGP=\{TP, TDO, IL, PR, DGO, DOD, SC, ME, ORU\}$  —  $i$ -я страница папки  $PGPi1$  гиперпроекта  $GP$ , состоящая из множеств (может быть и пустых), называемых текстовые поля  $TP$ , текстовые динамические окна  $TDO$ , рисунки (графики, чертежи и т.д.)  $IL$ , процессы  $PR$ , динамические графические окна  $DGO$ , диалоговые окна ввода-вывода данных  $DOD$ , сценарии  $SC$ , меню  $ME$ , органы управления  $ORU$ .

Известно, что если технологическое оборудование (или ТМ) можно представить в виде двух взаимодействующих частей — устройства управления (УУ) и объекта управления, то эти части проектируют различные службы и специалисты [2]. Проектирование УУ состоит из нескольких этапов : алгоритмического (создание алгоритмов управления и функционирования), логического (разрабатывается логическая структура УУ) и технического (схемотехнического).

В условиях всё усложняющихся УУ, использование программируемых контроллеров (ПК) всё больший удельный вес приобретают первые два этапа. Например, в [3] приводится таблица экспериментальных данных, свидетельствующая о том, что на них затрачивается до половины трудоёмкости на создание средств управления. При этом ошибки, допускаемые проектировщиками УУ на этих этапах, имеют принципиальный характер и более тяжёлые последствия, чем ошибки технической реализации. Поэтому всё более важное место занимает процесс проверки алгоритмов управления (АУ), являющихся фактически формализованным техническим заданием на проектирование УУ, в рамках системы моделирования.

В работах [4,5] излагаются основы построения комплексной САПР дискретных УУ технологическим оборудованием (ТМ) и процессами и рассматриваются вопросы использования мультимедийной среды ЭВМ для разработки этой САПР, которая получила название «Гиперсистема». В «Гиперсистеме» выполняется алгоритмическое (на уровне АУ), логическое (на уровне логической структуры) и схемное (на уровне принципиальной электрической схемы) моделирование проектов УУ ТМ и процессами.

В предлагаемой работе уделено внимание моделированию АУ ТМ на уровне их логического описания.

## 1. Моделирование АУ ТМ в виде СПФ

Рассмотрим в качестве описаний логических структур АУ ТМ систему предикатных формул (СПФ), которая отличается от систем логических уравнений (СЛУ) двумя существенными моментами, отмеченными в [6]:

в СПФ используются многозначные переменные, что существенно расширяет класс описываемых объектов;

при построении предикатных формул кроме логических операций И, ИЛИ, НЕ используются операции сравнения, что существенно расширяет класс описываемых ситуаций в управлении.

Моделирование логических описаний АУ ТМ может осуществляться с помощью так называемого метода сплошного моделирования[7]. Для существенного сокращения времени моделирования логических структур АУ, при возрастании их размерности, используют принцип событийности [7,8].

При изложении методов моделирования логических структур АУ в разных литературных источниках используются различные термины, понятия и буквенные обозначения. Для удобства изложения используем единую систему

обозначений и понятий, описанную в [8]. Отличительной особенностью используемой системы понятий является явное выделение в логическом описании (ЛО) АУ структурного и поведенческого аспектов.

Пусть имеем некоторый АУ. АУ назовем такое отображение

$S : A(X) \times B(Z) \rightarrow Z$ , в котором

1)  $S = \{S_1, S_2, \dots, S_j, \dots, S_J\}$ , где  $J$  – мощность множества  $S$ , определяющая некоторое конечное число предикатных формул в  $S$ ;

2)  $S_j : a_j(X) \times b_j(Z) \rightarrow z_j$  – некоторая  $j$ -я формула из  $S$ , она соответствует описанию одного выхода АУ;

3)  $X$  и  $Z$  есть некоторые конечные непустые множества логических переменных, входных и выходных соответственно;

4)  $A(X)$  есть элемент  $\{A(X)\}$  и является некоторым набором переменных из множества всех наборов переменных из  $X$ ;

5)  $B(Z)$  есть элемент  $\{B(Z)\}$  и является некоторым набором выходных переменных из множества всех наборов переменных из  $Z$ ;

6)  $a_j(X)$  и  $b_j(Z)$  есть наборы входных и выходных переменных соответственно для  $j$ -й формулы, причем  $a_j(X) \in \{A(X)\}$ ,  $b_j(Z) \in \{B(Z)\}$ ,  $z_j \in Z$ ; для  $\forall (a, b) \text{ I } (a \in a_j(X), b \in b_j(Z)) [ a \text{ q } b ]$ , где  $q \in \{\&, \text{I}, @, =, \neq, >, \geq, <, \leq\}$  есть символ логической операции или операции сравнения,  $a, b$  – многозначная или булева переменные либо их инверсии, т.е. любые переменные, являющиеся аргументами некоторой формулы  $S_j$ .

Структурная модель АУ может быть представлена также в скобочной форме.

Определение 2. Входной переменной  $x \in X$  логического описания  $S$  некоторого АУ будем называть такую многозначную переменную, значение которой не зависит от состояния АУ, т.е.  $(\forall x, x \in X) [x \notin Z \ \& \ X \cap Z = \emptyset]$ .

Соответственно, выходной переменной  $z \in Z$  логического описания  $S$  назовем такую переменную, значение которой определяется входными воздействиями  $A(X)$ , наборами выходных переменных  $B(Z)$  и некоторой формулой  $S_j$ , т.е.

$(\forall z, z \in Z) [S_j : a_j(X) \times b_j(Z) \rightarrow z]$ .

Определение 3. Состоянием входной  $x$  (или выходной  $z$ ) переменной будем считать текущее значение указанной переменной из множества  $X$  (или множества  $Z$ ) значений, причем  $X \in \{1, 2, \dots\}$ , а  $Z \in \{0, 1\}$ . Состояния переменных (входных и выходных) будем обозначать теми же символами, что и соответствующие переменные, но с индексами. Изменения состояний входных переменных представляют собой входные воздействия.

Определение 4. Шагом моделирования логического описания  $S$  назовем отрезок времени  $\vartheta$  между поступлениями на вход СПФ двух соседних воздействий  $A_i(X)$  и  $A_{i+1}(X)$ , или  $(\forall \tau, \tau \in \vartheta \mid A_i(X) \rightarrow C_i(Z)) [A_{i+1}(X) \rightarrow C_{i+1}(Z)]$ ,

где  $i = 0, 1, 2, \dots$  – номера входных воздействий и наборов состояний выходных переменных;  $C_i(Z)$  – набор состояний выходных переменных, соответствующий  $i$ -му входному воздействию.

Определение 5. Тактом моделирования логического описания  $S$  будем называть отрезок времени  $T$ , в течение которого изменяет свое значение хотя бы одна выходная переменная  $z \sim Z$ , или  $(@ T) [\# z, z \sim Z \ \& \ zT_k \neq zT_{k+1}]$ , где  $k \sim \{1, 2, 3, \dots\}$  – номер текущего такта моделирования;  $zT_k$  – состояние некоторой выходной переменной после окончания  $k$ -го такта.

С учетом введенных понятий состояний переменных, шага и такта моделирования дадим определение поведенческой модели.

Определение 6. Поведенческой моделью АУ назовем такое отображение

$F : A_i(X) \times B_{i,k-1}(Z) \rightarrow C_{i,k}(Z)$ , что

1)  $F = \{f_1, f_2, \dots, f_i, \dots, f_N\}$  – множество предикатных формул в модели  $F$ ;  $N$  – мощность указанного множества;

2)  $f_j : a_i(X) \times b_{i,k-1}(Z) \rightarrow Z_{i,k}$  – описывает последовательное изменение состояний  $j$ -го элемента АУ, где  $j \sim \{1, 2, 3, \dots\}$  – номер описываемого элемента;  $(i, k) \sim \{1, 2, 3, \dots\}$ ;  $i$  – номер шага и  $k$  – номер такта соответственно;

3)  $A_i \sim \{A(X)\}$  – входное воздействие для  $i$ -го шага моделирования из множества всех входных воздействий;

4)  $B_{i,k-1}(Z) \sim \{B(Z)\}$  – является набором состояний тех выходных переменных, значения которых из  $(k-1)$ -го такта участвуют в формировании набора состояний выходов на  $k$ -ом такте;

5)  $C_{i,k}(Z) \sim \{Z\}$  – набор состояний выходных переменных для  $i$ -го шага и  $k$ -го такта моделирования из множества всех наборов состояний выходных переменных  $Z$ ;

6)  $a_i(X)$ ,  $b_{i,k-1}(Z)$  и  $Z_{i,k}$  – входное воздействие, набор состояний выходных переменных и состояние выходной переменной соответственно для  $i$ -го шага,  $(k-1)$ -го и  $k$ -го такта моделирования.

Будем различать устойчивые и неустойчивые состояния алгоритмов управления ТМ, устойчивые и неустойчивые АУ.

Определение 7. Устойчивым состоянием некоторого АУ, функционирование которого описывается формулой  $F$ , называется такое состояние  $C_{i+1,k+p}(Z)$ , которое достигается из некоторого состояния  $C_{i,k}$  при подаче нового входного воздействия  $A_{i+1}(X)$  за конечное число тактов  $P$ .

Устойчивым назовем такой АУ, который в течение  $(i+1)$ -го шага моделирования за  $P$  тактов переходит в устойчивое состояние  $C_{i+1,k+p}(Z)$  из состояния  $C_{i,k}$ .

Начальному состоянию модели  $F$  соответствуют наборы  $A_0(X)$ ,  $B_0(Z)$  и  $C_0(Z)$ . Введем понятия ранга и множества рангов.

Пусть в СПФ некоторого АУ имеется множество  $\{f_{j1}\}$  таких ПФ, что множество наборов из  $\{a_{j1}(X)\}$  есть  $A_{j1}(X) \sim X$ , т.е.  $\{f_{j1} / j=1, 2, \dots \ \& \ \{f_{j1}\} \rightarrow \{a_{j1}(X)\}\}$ . Придадим такому множеству ПФ вес, равный 1, обозначив его через  $F_{j1,1}$ . Тогда множество формул  $\{f_{j2} / j_2= 1, 2, \dots \ \& \ \{f_{j2}\} \rightarrow (\{a_{j1}(X)\} \ \& \ F_{j2,2})\} = F_{j2,2}$  будет иметь вес 2.

Следовательно, некоторому  $j_q$ -му множеству ПФ такому, что

$$\{f_{jq} / jq = 0, 1, 2, \dots \ \& \ \{f_{jq}\} \rightarrow (\{a_{jq}(X)\} \ \& \ F_{j(q-1),q-1})\} = F_{jq,q},$$

будет соответствовать вес  $q$ .

Определение 8. Рангом  $r$  некоторого множества  $F_{jq,q}$  предикатных формул назовем численное значение веса  $q$  ( $q = 1, 2, \dots$ ), т.е.

$$(@ \ fjq \sim Fjq / q = 1, 2, \dots) [ \# \ r / r = !q!].$$

Объединение рангов  $r_k$  при изменении  $k$  от  $k=1$  до  $K$ , образует множество  $R$  рангов ПФ некоторой рассматриваемой СПФ, где  $K$  – мощность этого множества.

$$F ; A_i(X) \times B_{i,k-1}(Z) \rightarrow C_{i,k}(Z) . (1)$$

Предлагаемый способ, который описан в [8], назван способом динамического ранжирования. Здесь сделана попытка учесть недостатки статического ранжирования [9] и метода событийного моделирования. Суть динамического ранжирования состоит в реализации выборочного вычисления предикатных формул с помощью адресованных переходов. Моделирование здесь рассматривается как преобразование структурной модели в поведенческую модель. Для управления данным преобразованием используется матрица адресованных переходов.

Входные воздействия, поступающие извне на СПФ, активизируют формулы (т.е. могут привести к изменению значения формул), в которых соответствующие входные переменные являются аргументами. Активизированные формулы могут, в свою очередь, активизировать другие формулы, по отношению к которым они являются аргументами. Не исключена рекурсия, т.е. активизация формулы самой формулой (это имеет место для формул, описывающих элементы памяти).

Процесс активизации формул можно представить в виде дерева, каждый ярус которого соответствует одному этапу активизации. Номер яруса дерева активизации характеризует относительную упорядоченность формул и рассматривается нами как динамический ранг формулы, в отличие от общепринятого понятия ранга [7,8].

Пусть процесс активизации формул  $S_j$ , являющихся элементами системы  $S$ , представим в виде некоторого дерева  $G$ . Тогда вершину данного дерева обозначим через  $g_{k,l}$ , где  $(k,l) \in \{1,2,3,\dots\}$ ,  $k$  есть номер яруса дерева  $G$ ,  $l$  – номер вершины на  $k$ -м ярусе.

Определение 9. Дерево  $G$  назовем активизирующим графом, если:

- а)  $g_{k,l}$  является вершиной данного дерева;  $l$  принадлежит  $L$  – номер вершины  $k$ -го яруса,  $L$  – конечное число вершин на  $k$ -м ярусе;
- б) имеется формула  $w : g_{k,l} \rightarrow S_{jk}$ , которая сопоставляет каждой вершине графа  $G$  переход в активное состояние (т.е. изменение состояния) некоторой формулы  $S_{jk}$  на  $k$ -м шаге моделирования.

Теперь введем понятие динамического ранга.

Определение 10. Динамическим рангом ( $D$ -рангом) некоторой формулы  $S_j$  будем называть число, равное номеру  $k$  яруса активизирующего графа  $G$ , или  $(@ k, k \sim K)[\# D / D = k]$ ; где  $K$  – множество номеров ярусов активизирующего графа  $G$ .

Как следует из определения 10,  $D$ -ранг формулы изменяется в зависимости от того, какие входные воздействия поданы из СПФ (они соответствуют номеру такта моделирования  $k$ ) и какие формулы активизированы на предыдущем ярусе активизирующего графа  $G$ .

В нашем способе на каждом такте моделирования осуществляется определение активизированных формул и моделирование (вычисление) значений этих формул. При таком подходе такт моделирования и  $D$ -ранг формулы, согласно определений 9 и 10 имеют одинаковые численные выражения. Это обстоятельство позволяет существенно упростить процедуры

алгоритмической и программной реализации разработанного способа моделирования.

Особенности данного способа состоят в следующем:

способ не требует анализа структуры СПФ с целью выявления и разрыва обратных связей; обеспечивает существенное сокращение времени моделирования для СПФ, описывающих АУ, так как для последних характерна слабая функциональная связность;

предложенный способ не накладывает ограничений на структуру СПФ, что обеспечивает высокую адаптивность его к изменениям в постановке задачи моделирования.

При использовании программируемых контроллеров и персональных компьютеров для реализации сложных алгоритмов дискретного управления оборудованием этот этап становится определяющим. Хорошо и детально проработанный алгоритм управления обеспечивает простоту и эффективность разработки управляющих программ и другой проектной документации.

Наряду с этим все неточности и ошибки в алгоритме управления тиражируются и усиливаются на последующих этапах проектирования, изготовления, монтажа и запуска в эксплуатацию.

Рассмотрим реализацию описанного выше способа.

## 2. Объектно-событийная реализация динамического ранжирования.

Рассмотрим исследованный в работах [6,8] язык предикатных формул. Управляющая программа на этом языке представляется в виде СПФ. Этот способ наиболее близок к таким языкам представления управляющих программ для программируемых контроллеров как язык булевой алгебры, язык программных команд (AWL), язык релейно-контактной символики (КОР), язык функциональных схем (FUP).

При проектировании бортовых систем спутников связи [2,10] СПФ рассматривается как логическое представление алгоритма управления и является базой (техническим заданием) для разработки принципиальных схем. В этом и ряде других случаев (управляющие программы сложных станков и автоматических линий) СПФ может состоять из нескольких тысяч предикатных формул. Например, для оценки эффективности разработанных в работе [6] средств моделирования были использованы СПФ, содержащие около 5000 предикатных формул.

Мы рассматриваем СПФ как общую платформу, взаимодействие которой с перечисленными выше языками оказывается простым и естественным. Более того, в настоящей работе предлагается функциональное расширение СПФ, включающее основные функции программируемых контроллеров, операции и отношения теории множеств. Реализация кванторов всеобщности и существования фактически обеспечивается используемой системой программирования на этапе интерпретации (компиляции). В связи с отмеченным излагаемый ниже подход и средства являются универсальными и особо ценными.

Основная идея реализации метода динамического ранжирования [5] при моделировании СПФ заключается в представлении каждой предикатной формулы в виде объекта. СПФ представляется как совокупность объектов, взаимодействующих на основе событий. При этом в качестве события выступает изменение значения (переключение) предикатной формулы.

Входные воздействия на СПФ также представляются в виде объектов.

Рассматриваемая в данной работе задача объектно-событийного моделирования управляющих программ состоит в:

- создании классов моделей объектов, составляющих управляющую программу (АУ);
- построении модели управляющей программы (АУ);
- проверке выходов управляющей программы по эталонным выходам алгоритма управления;
- формировании функционального отображения СПФ и его графическое представление в виде графа управления.

### 3.1. Построение объектных моделей СПФ

В объектно-событийной модели СПФ основными объектами являются:

- Объект – предикатная\_формула;
- Объект – входное\_воздействие (вход).

В объектно-событийной модели каждый из перечисленных объектов имеет совокупность свойств и сценарий поведения. Сценарии поведения разработаны нами с учетом возможности визуально-диалогового взаимодействия пользователя с объектно-событийной моделью.

Процесс построения объектной модели СПФ заключается в выборе класса модели объекта из библиотеки классов, размещении модели и параметризации (настройки параметров) конкретного экземпляра класса.

Рассмотрим процесс построения модели простой СПФ, описывающей комбинационную схему [8] (см. рис. 1).

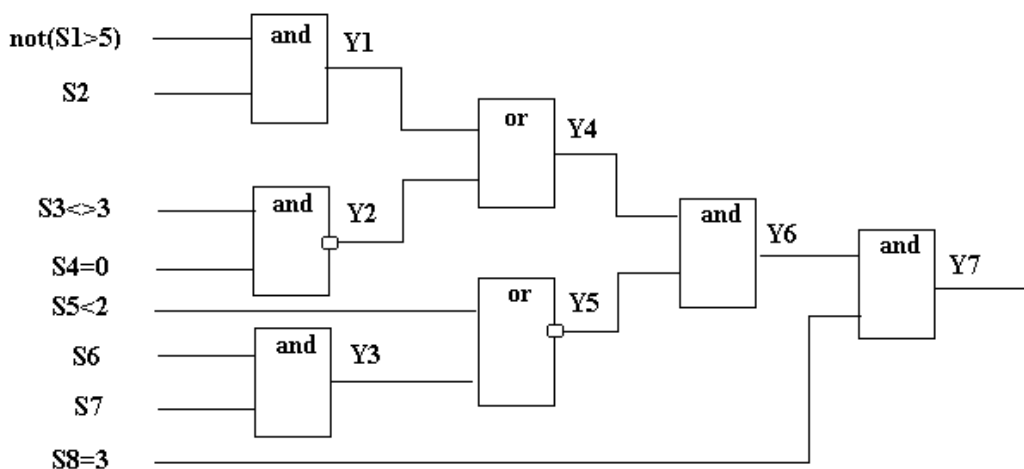


Рисунок. 1

На рис. 2 представлены входная последовательность, СПФ и соответствующие модели объектов процесса моделирования.

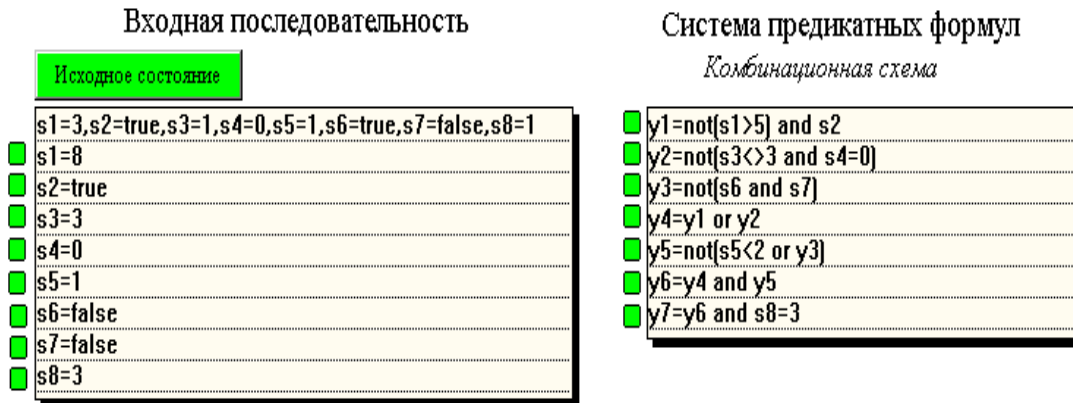


Рисунок. 2

Рассмотрим для примера упрощенные сценарии объекта “Вход”. Все необходимые пояснения сделаем в виде комментариев в тексте сценария.

```

-- *** Сценарий объекта “Вход” ***
To handle buttonclick
-- Получено сообщение о щелчке мыши по объекту
-- Это событие может создать пользователь
-- или любой другой объект
  Номер_такта_системы = 1
-- Номер такта системы (динамического ранга) равен 1
  s1=8
-- Входной переключатель s1 установлен в 8 положение
-- Для булевых входов используются значения True, False
  Действие=s1_8
-- В таком виде действие запишется в Графе управления
  send Записать_действие to field “Функциональная схема”
-- Посылаем сообщение Записать_действие в текстовое
-- поле “Функциональная схема”. Затем это действие
-- будет отображено в Графе управления
  send buttonclick to button "y1"
-- Посылаем сообщение buttonclick объекту (предикатной
-- формуле) “y1”, так как только в ней s1 - операнд
  fillcolor of self = red
-- После завершения моделирования на данном шаге
-- отмечаем поданный вход красным цветом
end
To handle rightbutttdown
-- Нажата правая клавиша мыши
  fillcolor of self = green
-- Отмечаем вход зеленым цветом перед
-- выполнением последующих воздействий на СПФ

```



end

Для полного понимания этих, а также приведенных в дальнейшем сценариев необходимо ознакомиться с [1,5]. Для построения модели СПФ пользователю такие знания не обязательны, так как приведенный выше и другие сценарии генерируются в приложении “Гиперсистема” автоматически. При этом пользователь задает только имена объектов, их действия и предикатные формулы.

#### 4. Моделирование СПФ

Основной целью моделирования СПФ является выявление и устранение неизбежных при проектировании сложных систем ошибок, имитация сбоев и отказов аппаратуры и оборудования, ошибки управляющего персонала (анализ по принципу “что - если”), а также получение наглядного, подробного и точного описания процесса работы. Более подробное изложение этой цели дано в [Мет пособие].

Моделирование начинается с воздействия пользователя на объект “Исходное состояние”. Соответствующая входная последовательность, СПФ и граф управления представлены на рис. 3.

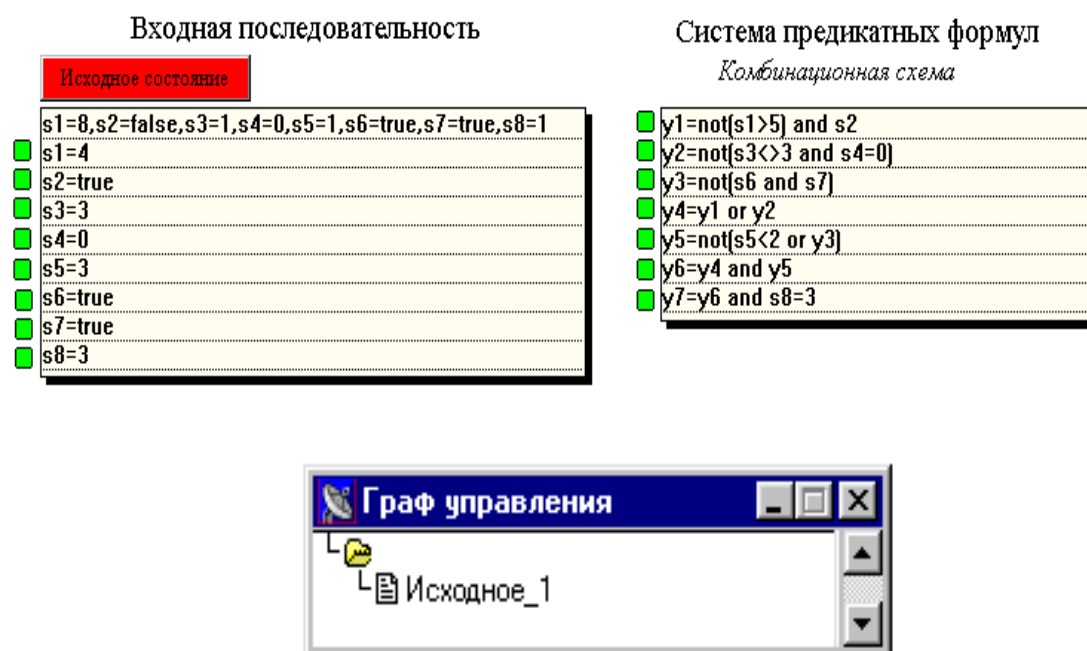


Рисунок. 3.

На рис. 4 представлены эти же данные в состоянии после подачи на СПФ отмеченных красным цветом входных воздействий. Последовательность подачи этих воздействий и последовательности переключений предикатных формул наглядно представлены на графе управления.

Следует напомнить, что для комбинационной схемы входные воздействия одной и той же входной последовательности могут подаваться в любом

порядке, вызывая одинаковое конечное состояние СПФ, однако порядок переключения предикатных формул в ответ на одно и тоже воздействие будет разным.

Этот факт, важный для анализа результатов моделирования сложных СПФ, становится ясным при сравнении двух графов управления на рис. 4.

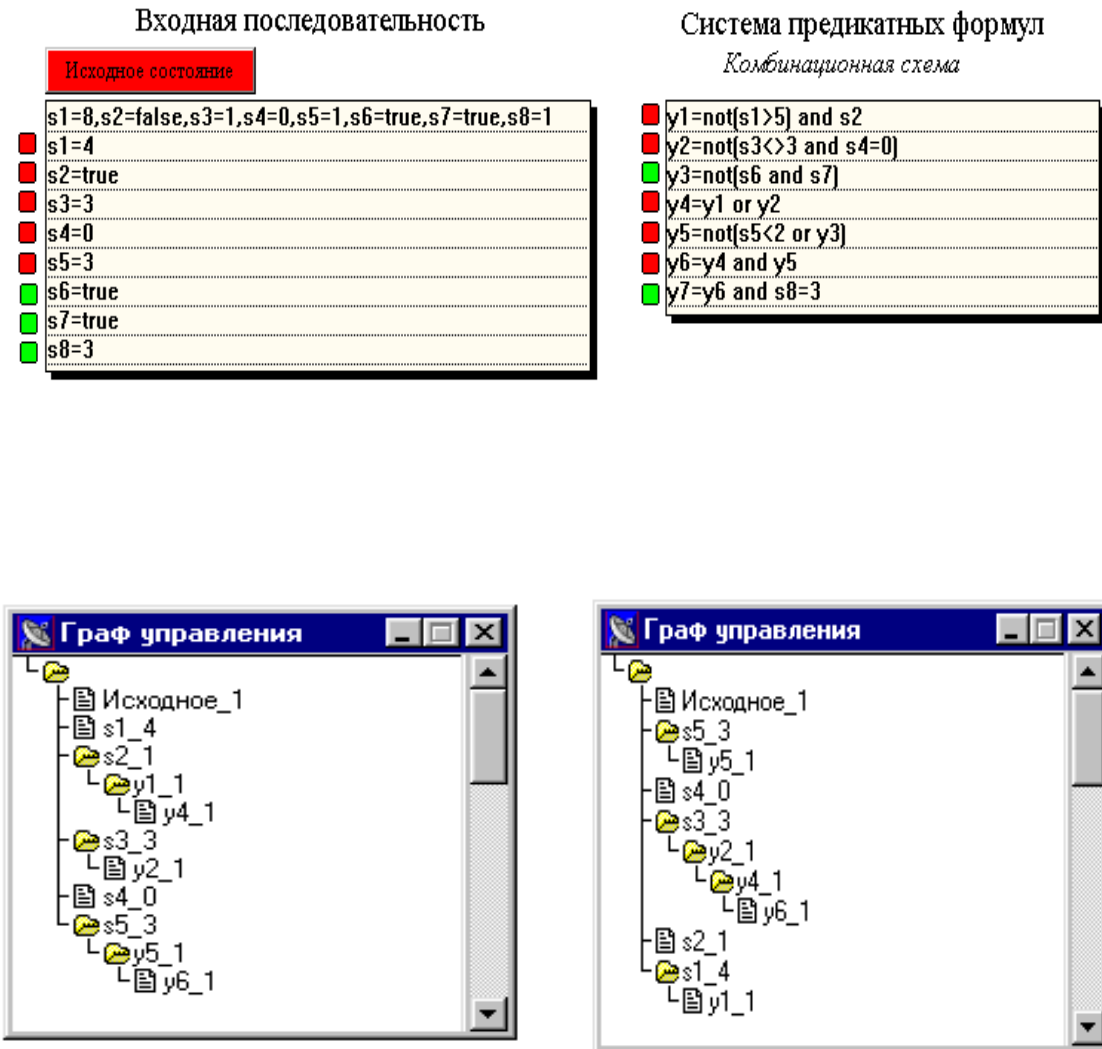


Рисунок 4.

На рис. 5 представлены результаты объектно – событийного моделирования СПФ для последовательностной схемы. При этом входные воздействия, записанные в поле “Входная последовательность”, могут произвольно изменяться пользователем как по порядку поступления, так и по значениям входных сигналов. Примером могут служить входы “s1” и “s6” (см. граф управления на рис.5).

## Входная последовательность

Исходное состояние	
	$s1=false, s2=false, s3=10, s4=false, s5=0, s6=0$
■	$s1=true$
■	$s2=true$
■	$s3=1$
■	$s4=true$
■	$s5=1$
■	$s6=5$

## Система предикатных формул

*Последовательностная схема*

■	$z1=s1 \text{ and } s2$
■	$z2=s4 \text{ and } s5=2$
■	$z3=\text{not}(s3>5 \text{ or } z1)$
■	$z4=\text{not}(s6<3 \text{ or } z2)$
■	$z5=\text{not}(z3 \text{ or } z6)$
■	$z6=\text{not}(z4 \text{ or } z5)$

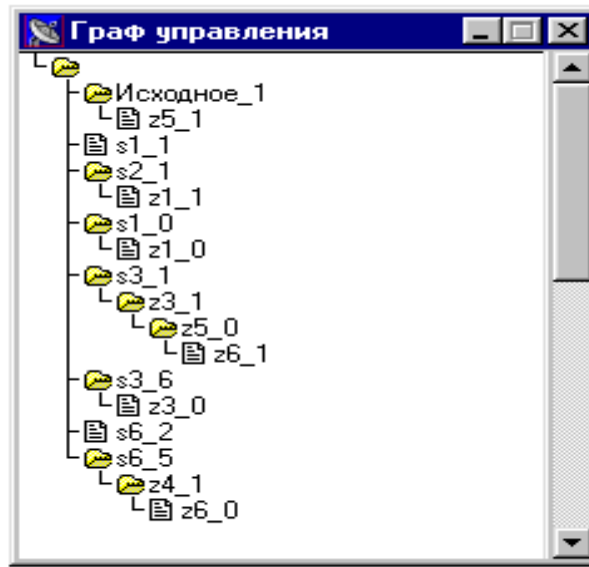


Рисунок 5.

На рис. 6 представлены результаты моделирования управляющей программы моечной машины в начале цикла работы. В СПФ этого примера триггеры МРМС, МР1Р, МРМО, МГМО и МРС упрощенно представлены логическими формулами, хотя более адекватно их поведение можно представить специальными функциями, имеющимися в составе вышеупомянутого функционального расширения.

Список предикатных формул  
Управляющая программа мойки

Исходное состояние

■	$NG = \text{not } SBA \text{ and not } S1 \text{ and not } S2 \text{ and } (\text{not } S3 \text{ or } NHP) \text{ and } S5$
■	$NHP = (S1 \text{ or } NHP) \text{ -- and } THP$
■	$THP = NHP$
■	$NAMO = \text{not } S1 \text{ and } S2 \text{ and not } S3$
■	$HCFZ = \text{not } S1 \text{ and } (\text{not } S2 \text{ or } NHP) \text{ and not } S3$
■	$HCFK = NAP$
■	$NAP = (SB1 \text{ and } (NG \text{ or } SB5) \text{ or } NAP) \text{ and } (\text{not } S4 \text{ or not } S5) \text{ and } (MPC \text{ or } KMMH) \text{ and } NAMO \text{ and not } SB4$
■	$KMMC = NAP \text{ and not } MPMC \text{ and } MPC$
■	$MPMC = ((KMMC \text{ and } S4) \text{ or } MPMC) \text{ and not } (\text{not } MPC \text{ or } SB4)$
■	$KMMB = NAP \text{ and } (KMMC \text{ and not } MPMC \text{ or } KMB \text{ and } KMS \text{ or } KMGB \text{ and not } MPC)$
■	$KMB = NAP \text{ and } MPMC \text{ and } (\text{not } MP1P \text{ and not } TOP \text{ or } (MPM0 \text{ or } MGM0))$
■	$TOP = (KMB \text{ or } TOP) \text{ and not } MP1P$
■	$KMMH = NAP \text{ and } (KMB \text{ and } MP1P \text{ or } MPM0 \text{ or } MGM0 \text{ or } MPC) \text{ and not } S5$
■	$MP1P = ((TOP \text{ and } KMMH \text{ and } S5) \text{ or } MP1P) \text{ and not } (\text{not } MPC \text{ or } SB4)$
■	$MPM0 = ((SB2 \text{ and } MP1P) \text{ or } MPM0) \text{ and not } (\text{not } MPC \text{ or } SB4)$
■	$KMS = NAP \text{ and } (MPM0 \text{ or } MGM0)$
■	$MGM0 = ((SB3 \text{ and } MP1P \text{ and not } MPM0 \text{ or } S5) \text{ or } MGM0) \text{ and not } (\text{not } MPC \text{ or } SB4)$
■	$KMOC = NAP \text{ and } MGM0 \text{ and } S4$
■	$TOC = KMOC$
■	$KMGB = NAP \text{ and } (S5 \text{ or } KMGB) \text{ and not } MPC$
■	$MPC = (S5 \text{ or } SB4 \text{ or } MPC) \text{ and not } (KMGB \text{ and } KMMB \text{ and } S4)$

Входы

- S1
- S2
- S3
- S4
- S5
  
- SBA
- SB1
- SB2
- SB3
- SB4
- SB5

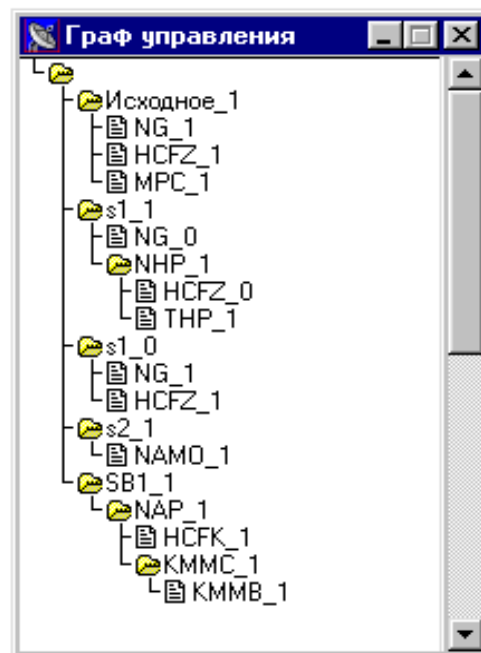


Рисунок 6.

В заключение отметим, что функциональные отображения СПФ в виде графов управления – это важнейший инструмент анализа сложных управляющих программ на всех этапах их жизненного цикла, включая проектирование, наладку, эксплуатацию и модернизацию. В самом деле,

обычный умозрительный анализ даже простейших рассмотренных СПФ вызывает существенные затруднения. Получение таких отображений даже для систем средней сложности вручную практически невозможно.

## ЛИТЕРАТУРА

1. Акунович С.И. Инструментальная система ToolBook для разработки мультимедийных приложений. – В кн. Основы построения комплексной САПР систем дискретного управления технологическим оборудованием и процессами./Уч.пособие в 2-х частях. Ч.1. Минск, изд-во БГТУ, 2000 г. - 42с. .

2. Добролюбов А.И., Акунович С.И. Автоматизация проектирования систем управления технологических машин. – М.: Машиностроение, 1974. – 224 с..

3. Калентьев А.А. Автоматизация проектирования алгоритмов управления технологическими системами с множеством дискретных состояний./Автореф.диссертации на соиск. уч. степени д.т.н. (на правах рукописи). - Самара, 1998. - 34 с.

4. Акунович С.И., Прохоренко Д.М. Моделирование проектов в приложении «Гиперсистема». – «Техника. Экономика. Организация», No.2, 1999. – с. 26-29.

5. Акунович С.И., Гончаров А.А., Дятко А.А. Моделирование и анализ проектов систем дискретного управления. – В кн. Основы построения комплексной САПР систем дискретного управления технологическим оборудованием и процессами./Уч.пособие в 2-х частях. – Ч. 2. – Минск, изд-во БГТУ, 2002. – 38 с.

6. Акунович С.И., Гончаров А.А. Описание алгоритмов управления технологических машин системой предикатных формул. – В сб. Применение систем автоматизированного электропривода на промышленных предприятиях Белоруссии./Тезисы докл. респ. научн. техн. совещания. – Мн.: БелНИИНТИ, 1982. – с. 91-92.

7. . Иоффе М.И. Программы сплошного и событийного моделирования в автоматизированной системе синтеза тестов цифровых схем./Тр. ин-та. Вып. 72. – М.: ИНЭУМ. 1977. –с.15-22

8. Гончаров А.А.Об одном подходе к моделированию логической структуры управляющих устройств// Информатизация образовательных процессов: автоматизация управления, технологии, дистанционное обучение: Сб. науч. ст.; Под ред. проф. Н.А.Цырельчука; В 2-х ч.(По итогам работы МНПК; Минск, 19-20 апр. 2001г.)/ Мин-во образования РБ. Минск. гос. высш. радиотехн. колледж.—Минск, 2001г.—Ч.1.—С.229-241

9. Гурвич Е.И., Козмин Л.Е. Ранжирование модели при логическом моделировании цифровых автоматов. – В сб. Автоматизация проектирования РЭА./ Мат-лы семинара.- М.: МДНТП, 1973. – с. 151-158.

10. Руководящий технический материал «Аппаратура радиоэлектронная бортовая. Типовая форма технических требований к радиоэлектронной аппаратуре в части выполнения логических операций ( РТМ 154-13-83 ). – Красноярск: НПО ПМ, 1983.

11. Гончаров А.А. О методе динамического ранжирования для моделирования систем логических уравнений. – В сб. Пути технического совершенствования и интенсификации разработки, производства и эксплуатации средств радиоэлектроники, электротехники и связи. /Тезисы докл.науч.-техн. конф.- Мн.: БелНИИНТИ, 1979. – с. 87-88.

## **Литература**