

УДК 004.021

**С. И. Акунович**, кандидат технических наук, доцент (БГТУ)**ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ  
ТЕХНОЛОГИЧЕСКИМ ОБОРУДОВАНИЕМ И ПРОЦЕССАМИ**

В статье рассмотрена интегрированная совокупность средств визуального моделирования систем логического управления на алгоритмическом и логическом уровнях их проектирования. Для моделирования в среде электронных таблиц Excel автором создано специальное логическое расширение. Для комплексного моделирования систем логического управления совместно с исполнительными механизмами предложены средства SCADA iFix.

In the article integrated set of means of visual modelling of systems of logic management at algorithmic and logic levels of their designing is offered. For modelling in the environment of spreadsheets Excel the author creates special logic expansion. For complex modelling of systems of logic management together with executive mechanisms means SCADA iFix are offered.

**Введение.** Современные технологические машины и процессы содержат многочисленные датчики, сигнализирующие об их состояниях. Сигналы с датчиков поступают на вход системы управления, которая должна анализировать текущую ситуацию и вырабатывать адекватные воздействия на исполнительные компоненты оборудования, непосредственно управляющие исполнительными механизмами: переключателями, клапанами, двигателями, насосами, электромоторами. Как сигналы с датчиков, так и управляющие воздействия дискретны, а чаще всего двоичные, т. е. управление становится логическим, и в этом случае системы управления принято называть системами логического управления.

Системы логического управления находят широкое применение не только на производстве, но и на транспорте, в атомной энергетике, сетях связи, информационно-справочных системах, ракетно-космической отрасли, компьютерах и бытовой аппаратуре и т. д.

Приступая к проектированию системы логического управления, нужно прежде описать алгоритм решения поставленной перед системой задачи, т. е. алгоритм управления.

Моделирование любой управляющей системы всегда начинается с алгоритмического описания процесса управления. Лишь после четкого описания алгоритма, его проверки и отладки можно перейти к конструированию технической системы, реализующей этот алгоритм. Если в описании алгоритма допущены ошибки, дальнейшая его реализация может оказаться бессмысленной.

**Основная часть. Визуальные модели на базе электронных таблиц.** Одной из наиболее распространенных версий табличных процессоров является программа Microsoft Excel, в которой для различных типов вычислений имеется большое число встроенных функций. В состав встроенных функций Excel входят и логические функции, что позволяет широко использовать этот табличный процессор для решения логических задач.

Функция ЕСЛИ позволяет реализовать ветвящуюся алгоритмическую структуру. В Excel можно также моделировать функции памяти (триггеры), включив режим циклических вычислений.

Однако для моделирования сложных систем логического управления перечисленных функций недостаточно. Автором настоящей статьи разработаны функции листа Excel для логических вычислений, эквивалентные функциям языка VBA (XOR, EQV, IMP).

Ниже приведен исходный код функции ИсИли (Исключающее ИЛИ).

```
Function ИсИли(ParamArray Аргументы() As Variant) As Boolean
'Эквивалентна XOR
'Истина, если только один аргумент истина
'Значения аргументов: 1 0 True False
Dim i As Integer
Dim Число_единиц As Integer
Число_единиц = 0
For i = 0 To UBound(Аргументы)
    If Аргументы(i) = 1 Or Аргументы(i) = True
Then Число_единиц = Число_единиц + 1
    Next i
    If Число_единиц = 1 Then ИсИли = 1 Else
ИсИли = 0
End Function
```

Кроме этого, разработаны функции листа Excel для логических вычислений с применением передних и задних фронтов аргументов.

Ниже приведен исходный код функции ПФ (Передний фронт).

```
Function ПФ(Аргумент As Range) As Boolean
'Истина, если аргумент изменился 0 -> 1
'Значения аргумента: 1 0
Dim Текущее_состояние As Variant
Dim Предыдущее_состояние As Variant
Текущее_состояние = Аргумент.Text
Предыдущее_состояние = Аргумент.Comment.Text
If Текущее_состояние = 1 And Текущее_состояние
<> Предыдущее_состояние Then
```

```

ПФ = Текущее_состояние
Else
    ПФ = 0
End If
    Аргумент.Comment.Text = ПФ
End Function
    
```

Визуальное представление логической модели системы логического управления обеспечивается на структурном (логические связи компонентов системы) и поведенческом (причинно-следственные связи компонентов системы) уровнях (рис. 1, 2).

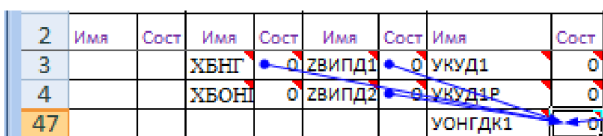


Рис. 1. Логические связи

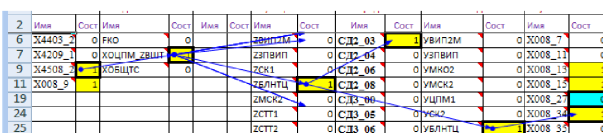


Рис. 2. Причинно-следственные связи

**Визуальные модели на базе средств промышленной автоматизации.** На этапах проектирования алгоритмов и программ управления широко используется язык последовательных функциональных схем (SFC). Это высокоуровневый графический язык, его основой является

математический аппарат двудольных графов – сети Петри.

Сети Петри (СП) применяются для анализа, моделирования и представления причинно-следственных связей в сложных распределенных дискретных системах, процессах параллельной обработки и синхронизации.

Описание СП может быть представлено в виде четверки  $G = (P, T, I, O)$  или графически в форме двудольных ориентированных графов (рис. 3), где

$P = \{P_1, \dots, P_n\}$  – множество вершин, называемых позициями;

$T = \{T_1, \dots, T_m\}$  – множество вершин, называемых переходами;

$I, O$  – соответственно входная и выходная функции (матрицы инцидентности).

Для описания динамических свойств сети (и, следовательно, моделируемой системы) вводится функция M-разметки сети (маркировки).

Язык SFC позволяет описать блок-схему программы – логику ее работы на уровне последовательных шагов и условных переходов; обеспечивает общую структуризацию и координацию функций управления последовательными процессами или машинами и механизмами [1, 2].

SFC-программа состоит из элементов двух типов: шагов (steps) и переходов (transitions), которые могут включать в себя элементы других языков.

На рис. 4 приведены условные графические обозначения основных компонентов SFC.

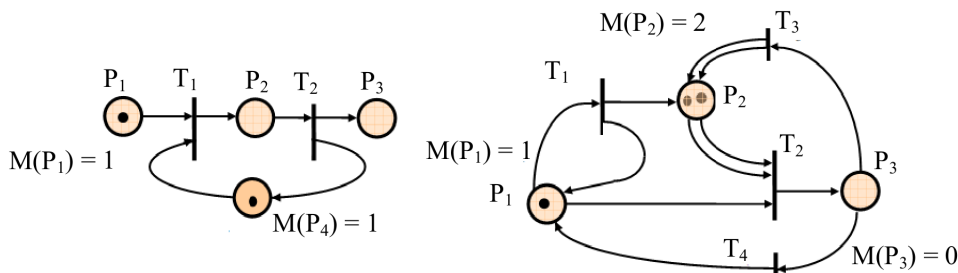


Рис. 3. Сеть Петри

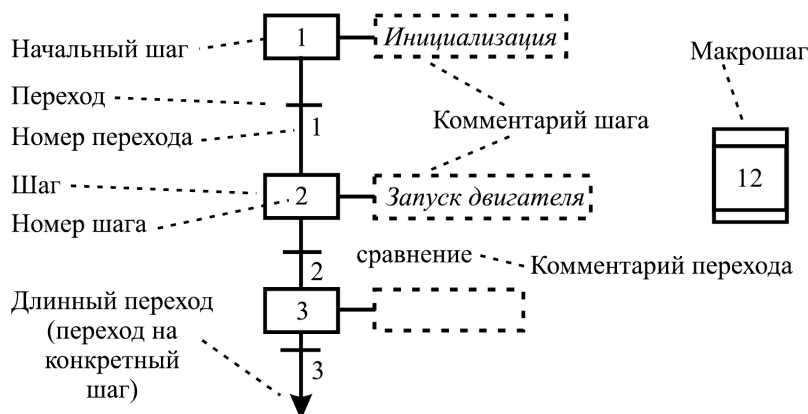


Рис. 4. Основные компоненты SFC

Язык релейных диаграмм (LD) применяется для описания логических выражений различного уровня сложности. Он содержит контакты (входные аргументы) и катушки (выходные переменные). Элементы организуются в сеть релейно-контактных схем.

Релейная схема представляет собой две вертикальные шины питания, между которыми расположены горизонтальные цепи из контактов и катушек реле (рис. 5).

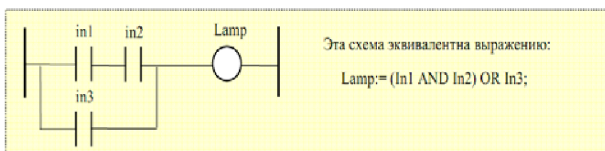


Рис. 5. Релейная схема LD

**Визуальные модели на базе языка UML [3].** В последнее десятилетие в компьютерном мире наметилась тенденция моделирования сложных программных систем визуальными (наглядны-

ми) моделями (рис. 6) на унифицированном языке моделирования (UML).

На этой схеме между предметными областями и программными кодами появились диаграммы языка UML и их математическая основа – теории множеств и графов.

Рис. 6 также показывает, что UML-диаграммы могут преобразовываться в исходный код (прямое преобразование) и, наоборот, исходный код может преобразовываться в диаграммы (обратное преобразование).

На рис. 7 представлены фрагмент диаграммы классов программного обеспечения процессорного модуля блока управления космического аппарата и фрагмент сгенерированного кода в среде DELPHI.

**Визуальные модели на базе языка ДРАКОН.** Язык разработан в космической промышленности совместными усилиями Российского космического агентства (НПЦ автоматики и приборостроения, г. Москва) и Института прикладной математики РАН имени М. В. Келдыша [4, 5].

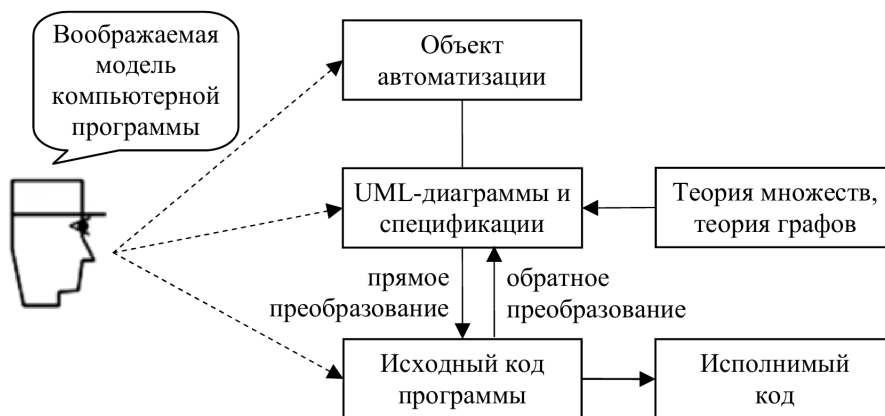


Рис. 6. UML-модель программной системы

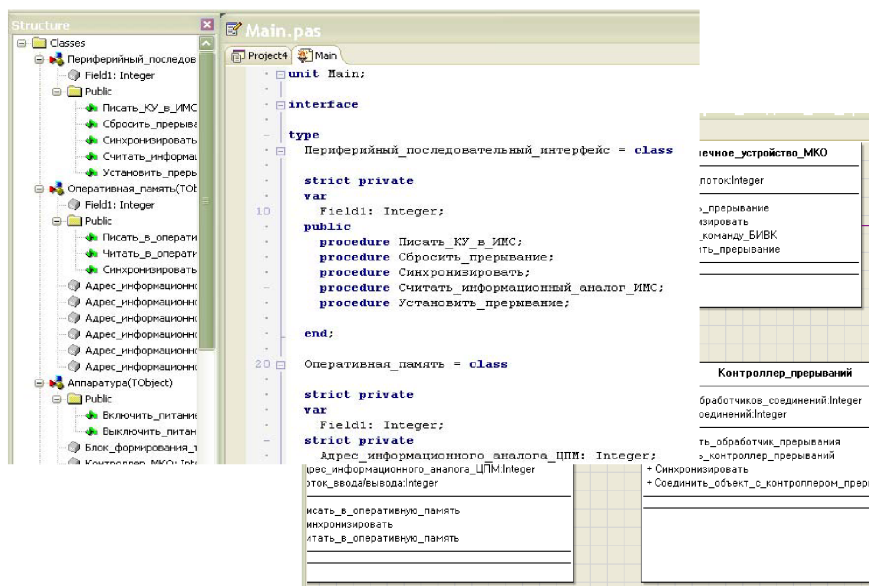


Рис. 7. Диаграмма классов и сгенерированный код DELPHI

Схемы и алгоритмы, разработанные с помощью этого языка, просты и понятны, благодаря чему разработку многих компьютерных программ для космических ракет на практике ведут не программисты, а обычные специалисты – по принципу «программирование без программистов».

Создано семейство языков программирования как оригинальных (ДРАКОН-2), так и гибридных (ДРАКОН-СИ, ДРАКОН-МОДУЛА, ДРАКОН-ПАСКАЛЬ, ДРАКОН-БЕЙСИК и т. д.) (рис. 8).

Ключевые слова LOOP, IF-THEN-ELSEIF-ELSE, EXIT, END	Программа на языке МОДУЛА-2 LOOP K := K + 2; N := N + 3; IF K > 10 THEN EXIT ELSEIF N > 20 THEN EXIT ELSE X := A + K + N; END	Программа на языке ДРАКОН-МОДУЛА 
Ключевые слова IF-THEN-ELSE, CASE-OF, OR, END, WRITELN	Программа на языке ПАСКАЛЬ IF ( K=1 OR K=2 ) THEN CASE K OF 1: X := SIN(X); 2: X := COS(X); END (* CASE *) ELSE WRITELN ("ОШИБКА");	Программа на языке ДРАКОН-2 

Рис. 8. ДРАКОН-МОДУЛА и ДРАКОН-ПАСКАЛЬ

**Визуальные комплексные модели на базе SCADA-систем.** Комплексное моделирование систем логического управления, включая датчики и исполнительные механизмы, наиболее эффективно может выполняться в SCADA-системах. В настоящей работе рассматривается HMI/SCADA iFix [6]. На рис. 9 дана общая схема управления автоматизированным процессом.

Визуальная анимированная модель автоматизированного процесса, включая исполнительные органы и компоненты оборудования, представлена на рис. 10.

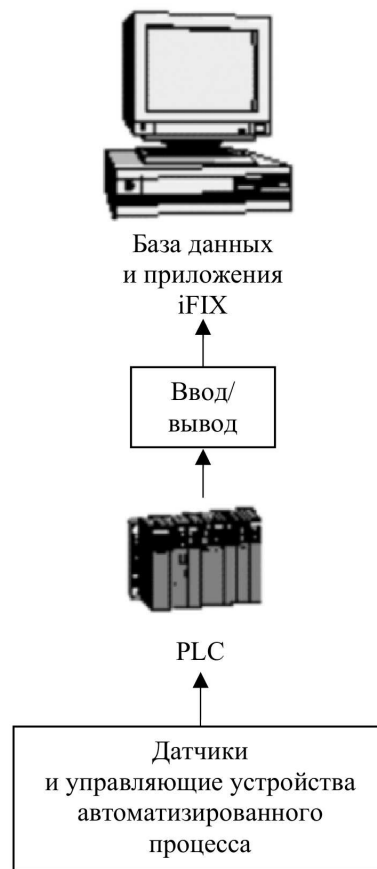


Рис. 9. Схема управления iFix

Рис. 11 иллюстрирует базу данных автоматизированного процесса на примере аналогового ввода.

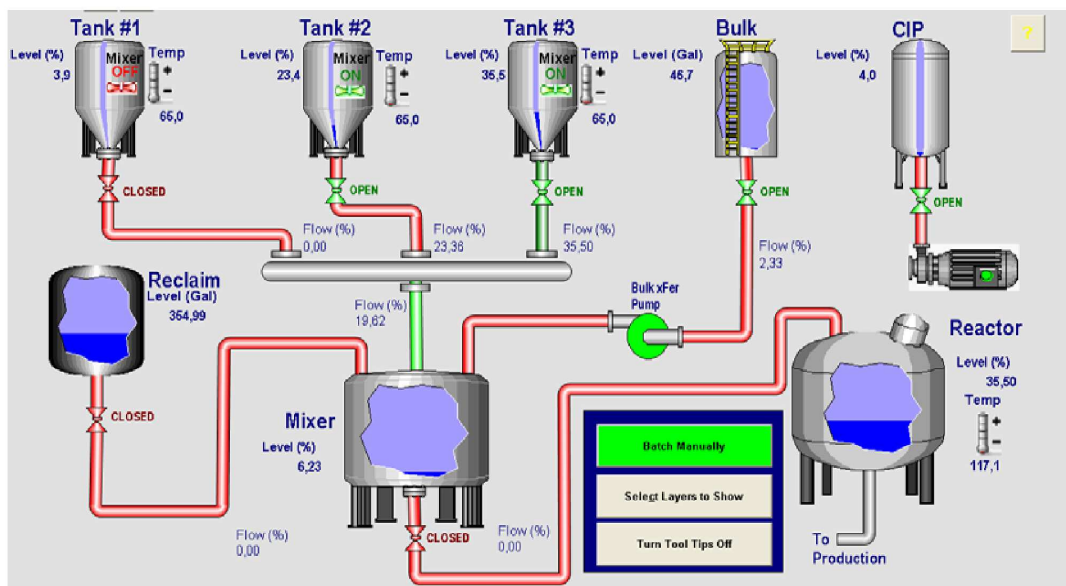


Рис. 10. Визуальная модель автоматизированного процесса

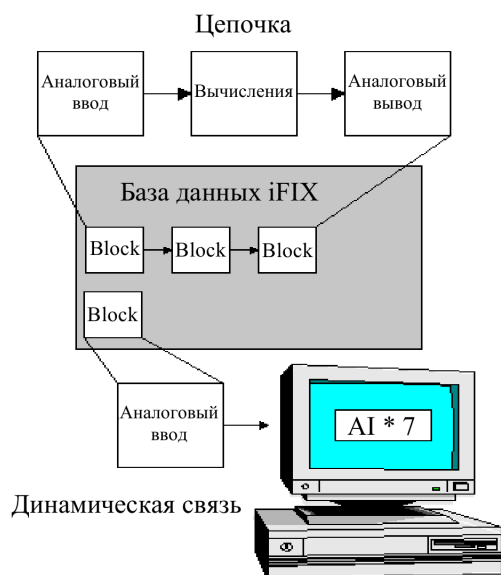


Рис. 11. База данных процесса

**Заключение.** На базе предложенной в данной работе интегрированной совокупности визуальных моделей разрабатывается вторая версия приложения «Гиперсистема» [1] для решения задач алгоритмического и логического анализа систем логического управления технологическими машинами и процессами.

#### Литература

1. Акунович, С. И. Дискретные системы логического управления технологических машин /

С. И. Акунович, А. А. Гончаров, Ю. Н. Петренко. – Минск: Юнипак, 2006. – 334 с.

2. Акунович, С. И. Об использовании визуального программирования при изучении алгоритмов управления дискретных устройств / С. И. Акунович // Информатизация образовательных процессов: автоматизация управления, технологии, дистанционное обучение: материалы Междунар. науч.-практ. конф., Минск, 19–20 апр. 2001 г. / МГВРК. – Минск, 2001. – С. 152–162.

3. Новиков, Ф. А. Моделирование на UML / Ф. А. Новиков, Д. Ю. Иванов. – СПб.: Наука и техника, 2010. – 640 с.

4. Паронджанов, В. Д. Как улучшить работу ума: Алгоритмы без программистов – это очень просто! / В. Д. Паронджанов. – М.: Дело, 2001. – 360 с.

5. Шамардина, Е. И. Язык программирования «ДРАКОН» и его применения за пределами ракетно-космических проектов. Разработка математической модели и редактора / Е. И. Шамардина, П. А. Манюнин // Актуальные проблемы российской космонавтики: труды XXXIV Академических чтений по космонавтике, Москва, январь 2010 г. – Москва, 2010. – С. 494–495.

6. Патрахин, В. Новое поколение HMI/SCADA-систем – Proficy iFix 5.0 RUS / В. Патрахин // Мир автоматизации. – № 1. – март 2010.

*Поступила 26.02.2011*