

УДК 681.3

И.А. Баранов (ПАО «ИНЭУМ им. И.С. Брука»)

I. Baranov

## ГЕНЕРАЦИЯ УПРАВЛЯЮЩЕГО КОДА НА ОСНОВЕ АНАЛИЗА P&ID СХЕМ

### CONTROL CODE GENERATION BASED ON P&ID DIAGRAM ANALYSIS

*В статье предлагается метод генерации управляющего кода для ПЛК в соответствии со стандартом IEC 61131-3 на основе анализа P&ID диаграмм. Рассматривается создание экспертной системы с помощью инструмента CLIPS. Приводится пример её использования для автоматизации генерации управляющего кода блокировок технологического объекта.*

*This paper describes the approach of IEC 61131-3 languages code generation based on P&ID diagram analysis. The process of creation expert system is under consideration. The paper also describes the example of interlock control code generation for plant using created expert system.*

*Ключевые слова: генерация кода, P&ID, МЭК 61131-3, ПЛК, CLIPS.*

*Keywords: code generation, P&ID, IEC 61131-3, PLC, CLIPS.*

#### Введение

В настоящее время разработка современных прикладных программ для ПЛК (Программируемых Логических Контроллеров) серии СМ1820М на базе микропроцессоров с архитектурой Эльбрус, SPARC, ARM, x86, работающих на нижнем уровне систем управления в области промышленной автоматизации, осуществляется с помощью среды разработки Veremiz, поддерживающей языки стандарта IEC 61131-3 (FBD, SFC, LD, ST, IL). Такие языки позволяют описывать необходимые алгоритмы с помощью высокоуровневых текстовых и графических абстракций. Среда разработки

Beremiz может исполняться под ОС Windows и ОС Эльбрус [1].

Одной из современных тенденций в области разработки программного обеспечения является активное использование генерации кода, позволяющей в конечном результате повысить скорость, качество и надёжность получаемого решения. В сфере промышленной автоматизации при реализации управления и мониторинга определённого технологического объекта требования могут быть представлены разными способами, в том числе и с помощью так называемых инженерных артефактов. Они могут описывать в графической и/или текстовой форме физическую структуру технологического объекта, связи между различными его компонентами и используемыми приборами, механизмы запуска/остановки системы, аварийные остановки, блокировки и т.д. Анализ этих данных может быть использован в автоматизации методов создания программных компонент, например управляющих прикладных программ для ПЛК. В частности, существуют определённые программные части, представленные на языках стандарта IEC 61131-3, которые подвержены постоянным ошибкам и повторяются из проекта в проект. Создание (генерация) таких компонент более высокоуровневыми программными средствами вместо ручной работы может заметно улучшить качество получаемых решений.

Вопрос о возможности анализа инженерных артефактов уже поднимался в ряде научных исследований. В статье [2] рассматривается генерация SFC и LD схем из диаграмм, определённых стандартом ISA-88. Работа [3] демонстрирует генерацию причинно-следственных таблиц и графических схем на языках LD и FBD, анализируя R&ID диаграммы. В результате прототипы созданных приложений, выполняющие генерацию управляющего кода для ПЛК, заметно увеличили качество создаваемых решений и показали практическую значимость предлагаемых методов.

В данной статье представлен пример реализации частичной автоматизации процесса разработки прикладных программ для ПЛК, основанный на генерации кода, получаемого в ходе исполнения экспертной системы (и других программных компонент),

анализирующей требования к системе управления, представленной в виде P&ID диаграмм (A piping and instrumentation diagram/drawing, диаграммы взаимосвязи технологического оборудования и приборов) – одного из самого распространенного инженерного артефакта для описания требований.

### Представление требований к системе управления с помощью P&ID диаграмм

В рамках исследования выбрано изучение возможности анализа P&ID диаграмм, применяемых как инженерный артефакт описания требований к разрабатываемой системе управления, и генерации управляющих алгоритмов на основе их описания. Фрагмент такой диаграммы продемонстрирован на рис. 1:

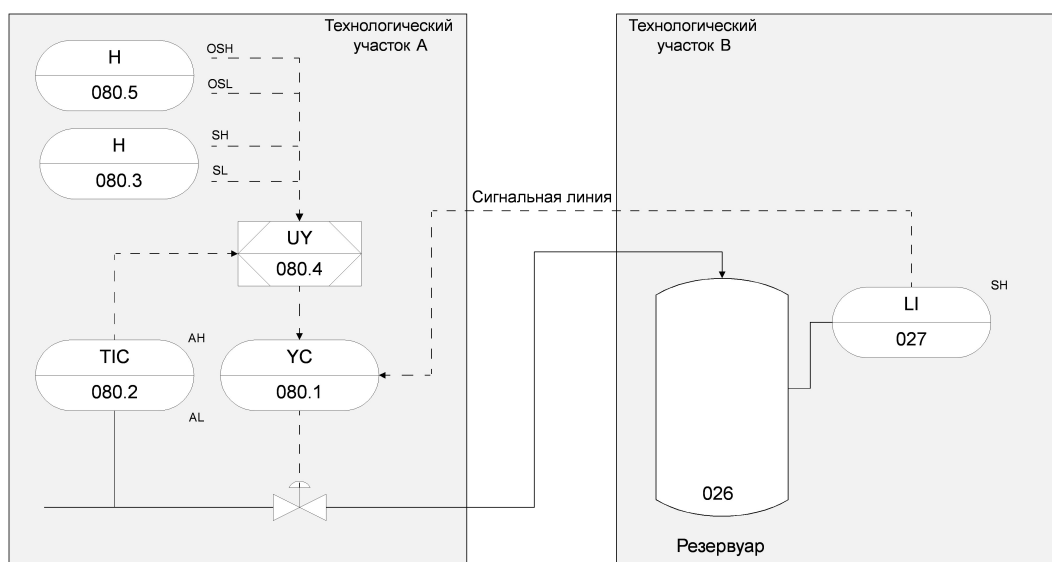


Рисунок 1. Фрагмент P&ID диаграммы технологического объекта

Диаграммы P&ID, дополненные рядом документов, содержат информацию, необходимую для точного определения (в рамках автоматизации технологического объекта) расположения требуемых датчиков, приборов и исполнительных механизмов, функций управления, механизмов поддержания системы в стационарном состоянии и гарантии её безопасной работы. Любая P&ID диаграмма содержит множество каналов связи (pipes) между технологическим оборудованием и приборами управления и мониторинга системы (например, между насосами, резервуарами, клапанами).

Графическое представление данных элементов определено стандартом ISO 10628. Классификация приборов осуществляется с помощью соответствующих кодов. Рассмотрим элемент диаграммы, отображаемый с помощью овала с поперечной линией, представленный на рис. 2:

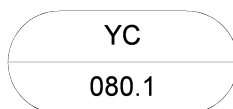


Рисунок 2. Элемент P&ID диаграммы

Из рис. 2 видно, что вверху алфавитный код задан значением УС. где первый символ У определяет функциональное назначение данного элемента как «actuation setting», т.е. приведение в действие клапана, а второй символ С означает «control», т.е. управление. Одинарная поперечная линия сообщает о том, что элемент находится в центральном пункте управления (отсутствие линии означает, например, что элемент монтируется на месте к трубам, резервуарам и другим элементам объекта управления). В результате данный элемент отвечает за управление клапаном, который находится ниже него на рис. 1. Все возможные комбинации алфавитных кодов имеют чёткое функциональное значение: L – уровень, Т – температура, Р – давление, А – сигнал тревоги, I – отображение аналогового значения и т.д. Ниже поперечной линии задан идентификационный номер 080.1 в рамках данной P&ID диаграммы. Также на диаграмме присутствует элемент с алфавитным кодом ТИС и идентификационным номером 080.2, измеряющий температуру в трубе и передающий соответствующий сигнал элементу с алфавитным кодом УУ и номером 080.4. Шестиугольник УУ представляет собой программный компонент, обрабатывающий значение температуры и принимающий на основе этого решение о передаче сигналов управляющему клапану 080.1 (между ними есть сигнальный интерфейс). Обозначения SH, SL, OSH и OSL вне овалов описывают механизмы сигнализации при наступлении некоторого события (например, превышение уставки). Символ S в самом начале соответствует необходимости реализации автоматического действия в ответ на

превышение или понижение ниже допустимого значения у технологического объекта, а символ O говорит о наличии определенной индикации наступившего события. Символы H и L в описании данных требований определяют уставки верхнего (High) и нижнего (Low) пределов соответственно.

Для представления P&ID диаграмм в электронном виде определён стандарт IEC 62424, носящий имя CAEX (Computer Aided Engineering Exchange) [4]. Описание P&ID диаграммы в формате CAEX содержит (и позволяет определить) собственные классы для оборудования и управляющих структур, а также различные их свойства (например, тип, интервал изменения и т.д.). Благодаря возможности представления P&ID диаграмм в формате CAEX, который синтаксически и семантически является форматом XML, появляется возможность реализации анализа такого представления сторонними приложениями и средствами.

### **Структура и основные компоненты предлагаемой системы**

Предлагаемая система дает возможность произвести анализ представления P&ID диаграммы, выявить на ней основные компоненты и связи между ними и в результате предложить определенное решение задачи автоматизации в виде управляющего кода на языках IEC 61131-3, основываясь на знаниях из данной предметной области. В рамках упрощения исследования рассмотрены следующие базовые компоненты: клапан, резервуар, насос, датчик уровня жидкости. Исполнение системы происходит следующими этапами:

- формализация основных элементов, участвующих в управлении технологическим процессом;
- получение и первоначальная обработка данных из формата CAEX;
- выполнение анализа полученных данных: выявление компонент технологического процесса и взаимосвязей между ними, а также необходимых управляющих алгоритмов;
- ввод фактов, формализованных на предыдущем этапе, в экспертную систему;
- выявление экспертной системой шаблонов технологических объектов и приборов на основе базы знаний по средствам выполнения правил;

- генерация управляющего кода на основе выполненных на предыдущем шаге правил;
- интеграция полученного кода на языках IEC 61131-3 в проекты, реализованные в среде Veremiz.

Получение и первоначальная обработка данных из формата CAEX выполняются с помощью языка Python и библиотек для работы с XML, встроенных в язык. Также для удобства работы созданы классы Valve, Vessel, Pipe, SignalConnection и другие на языке Python, соответствующие приборам, технологическому оборудованию, сигналам и т.д. В них реализованы программные интерфейсы для возможности переноса связей из CAEX формата P&ID диаграммы в программное представление.

Рассмотрим пример «разбора» одного из элементов P&ID диаграммы в формате CAEX, отвечающего за управление клапаном:

#### **Листинг 1. Структура определения факта в CLIPS**

```
<InternalElement localElementName="080.1">
  <RoleRequirements RoleClassLibName="IEC PASS6xxxx RoleLib"
refRoleClass="PCE request">
  <AdditionalAttribute LocalAttributeName="C processing function">
    <DefaultValue>true</DefaultValue>
  </AdditionalAttribute>
  <AdditionalAttributeValue refAttribute="PCE category">
    <Value>Y</Value>
  </AdditionalAttributeValue>
  <AdditionalAttributeValue refAttribute="Location">
    <Value>Central</Value>
  </AdditionalAttributeValue>
  ...
  <AdditionalExternalInterface LocalInterfaceName="Y"
refInterfaceClass="ActuatorSource"/>
  </RoleRequirements>
</InternalElement>
```

Значение атрибута «PCE category» равно «Y», следовательно, согласно стандарту CAEX, данная категория относится к приведению в действие клапана. Атрибут «LocalAttributeName» равен «C processing function» и относится к управлению. После анализа данных атрибутов создаётся объект соответствующего класса на языке Python. Данное действие выполняется для всех приборов, технологических объектов и связей между ними.

Реализация экспертной системы выполняется на основе известного инструмента CLIPS [5]. Главным преимуществом создания и использования экспертной системы перед вариантом разработки обычной программы является возможность чёткого разделения между текущими знаниями и механизмами получения новых из них. Тем самым упрощается ввод новых знаний, добавление новых правил вывода и т.д. [6].

В рамках данного исследования для «бесшовной» интеграции с различными шагами создаваемой системы будет использоваться библиотека PyCLIPS [7], позволяющая работать с CLIPS из языка Python. Данная возможность позволяет использовать объекты классов, созданные на предыдущем этапе, для ввода необходимой информации в экспертную систему. В инструменте CLIPS для создания экспертных систем используются следующие основные сущности: факты, правила и переменные. Описание фактов начинается с ключевого слова «assert»:

#### **Листинг 2.** Структура определения факта в CLIPS

```
(assert факт1)
(assert факт2)
(assert факт3)
```

Определение правила может быть задано следующим образом:

#### **Листинг 3.** Структура определения правила в CLIPS

```
(defrule rule1
  (шаблон1)
  (шаблон2)
  ...
=>
  (действие1)
  (действие2)
  ...
)
```

**В описании правила (Листинг 3):** до оператора => – определены условия, а после оператора => – заданы действия, исполняемые в случае выполнения условий, тем самым активирующие правила. Шаблон в виде факта может быть представлен более сложной структурой, где часть его параметризирована и представлена как переменная, которая далее используется в действии. Действие может содержать создание новых правил,

генерацию текстовых данных и т.д. После того как все правила и факты определены в CLIPS, вызывается метод `run()`, запускающий исполнение системы.

Как было сказано выше, в рамках исследования используется библиотека `PyCLIPS`, и в связи с тем, что все действия в рамках активации правил должны генерировать код, то в Python используется вызов `code = clips.StdoutStream.Read()` для получения текстового значения генерируемого кода. Далее этот текст используется для интеграции в проект, создаваемый в среде `Veremiz`.

## Пример

В качестве примера рассмотрим генерацию блокировок из P&ID диаграммы, отображающей некоторый участок технологического объекта. Блокировки являются важной частью системы управления и отвечают за корректное и безопасное её исполнение. На рис. 1 был представлен фрагмент P&ID диаграммы исследуемого технологического объекта. Представление данной P&ID диаграммы в формате CAEX выглядит следующим образом (приведён фрагмент):

### Листинг 4. Фрагмент CAEX представления P&ID диаграммы

```
<CAEXFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SystemUnitClassLib GlobalSystemUnitLibName="AnnexLib">
    <SystemUnitClass SystemUnitClassName="B.7">
      <InternalElement localElementName="080.1">
        <RoleRequirements refRoleClass="PCE request">
          <AdditionalAttribute LocalAttributeName="C processing function">
            <DefaultValue>true</DefaultValue>
          </AdditionalAttribute>
          <AdditionalAttributeValue refAttribute="PCE category">
            <Value>Y</Value>
          </AdditionalAttributeValue>
          ...
        </RoleRequirements>
      </InternalElement>
      <InternalElement localElementName="080.2">
        <RoleRequirements refRoleClass="PCE request">
          <AdditionalAttribute LocalAttributeName="I processing function">
            <DefaultValue>true</DefaultValue>
          </AdditionalAttribute>
          <AdditionalAttribute LocalAttributeName="C processing function">
            <DefaultValue>true</DefaultValue>
          </AdditionalAttribute>
          <AdditionalAttributeValue refAttribute="PCE category">
            <Value>T</Value>
          </AdditionalAttributeValue>
          ...
        </RoleRequirements>
      </InternalElement>
    </SystemUnitClass>
  </SystemUnitClassLib>
</CAEXFile>
```



```

<AdditionalExternalInterface LocalInterfaceName="TIC"
  refInterfaceClass="SignalSource"/>
...
  </RoleRequirements>
</InternalElement>
...
<InternalLink localLinkName="027LI_080.1YC" refPartnerSideA="027"
refPartnerInterfaceSideA="LI" refPartnerSideB="080.1"
refPartnerInterfaceSideB="YC"/>
...

```

Элемент LI отображен как Level Indicator (индикатор уровня), а YC является функцией управления клапаном. Между ними существует сигнальный интерфейс, определённый с помощью XML-тега с названием InternalLink.

Передадим в разработанную систему в качестве входного потока данных XML-файл с CAEX-описанием данной диаграммы. После исполнения первого этапа в экспертную систему будет введён ряд следующих фактов:

**Листинг 5.** Список добавленных после анализа P&ID диаграммы правил

```

(vessel 026)
(equipment high-level-indicator 027)
(equipment controlled-valve 080.1)
(signal-connection 027 080.1)
(signal-connection 080.1 026)
(control-function 080.4)
(pipe 080.1 026)

```

Данные факты представляют конкретное состояние знаний в системе. Они соответствуют программной реализации выполненного ранее анализа CAEX-представления. В экспертной системе также определён вручную ряд правил, позволяющих найти наличие технологического оборудования. Ниже приведен пример правила max\_level\_interlock в инструменте CLIPS, «ищущего» необходимость генерации блокировки подачи жидкости при достижении превышения уровня жидкости:

**Листинг 6.** Определение правила в CLIPS для генерации блокировки

```

(defrule max_level_interlock
(vessel ?vessel_tag_name)
(equipment high-level-indicator ?hli_tag_name)
(equipment controlled-valve ?cv_tag_name)
(pipe ?cv_tag_name ?vessel_tag_name)
(control-function ?cf_tag_name)
(signal-connection ?hli_tag_name ?cf_tag_name)
(signal-connection ?cf_tag_name ?cv_tag_name)

```

```

(signal-connection ?hli_tag_name ?vessel_tag_name)
=>
(printout t «PROGRAM Interlock1\nVAR\n  level_value_?vessel_tag_name:
REAL;\n  valve_value_?cv_tag_name: BOOL;\nEND_VAR\nVAR RETAIN\n
maximum_level_value: REAL;\nEND_VAR\n\nIF level_value_?vessel_tag_name >
maximum_level_value THEN\n  valve_value_?cv_tag_name := True;\nELSE\n
valve_value_?cv_tag_name := False;\nEND_IF\n\nEND_PROGRAM\n»)
)

```

В этом правиле в области условий используются факты наличия резервуара, прибора мониторинга уровня жидкости в резервуаре, сигнальных интерфейсов, трубы и необходимости программной реализации функции управления клапаном. Факт (vessel ? vessel\_tag\_name) представляет поиск соответствия в введённых ранее фактах резервуара с определённым идентификационным номером ?vessel\_tag\_name, который в рамках правила определён как переменная для использования в дальнейших сравнениях. Подобным образом определены факты наличия оборудования контроля уровня жидкости (equipment high-level-indicator ?hli\_tag\_name) с требованием контроля превышения уровня и управляемого клапана (equipment controlled-valve ?cv\_tag\_name). В условии (pipe ? cv\_tag\_name ?vessel\_tag\_name) идет поиск наличия связи в виде трубы между управляемым клапаном с идентификационным номером, определённым в переменной ? cv\_tag\_name, и резервуаром с ранее найденным номером ?vessel\_tag\_name. Факт (control-function ?cf\_tag\_name) определяет требование к программной реализации функции управления клапаном. Три условия signal-connection соответствуют наличию сигнальных связей между датчиком уровня жидкости и функцией управления, функцией управления и управляемым клапаном, а также датчиком уровня жидкости и резервуаром. После того как введены все факты в базу знаний экспертной системы и определены все правила, она готова к исполнению. Выполнение всех условий в правиле приводит к его «активации» и выдаче результата.

После выполнения экспертной системы получен управляющий код в текстовой форме на языке ST стандарта IEC 61131-3, который может быть интегрирован в проект, созданный в среде Veremiz. Далее приведен фрагмент сгенерированного кода.

### Листинг 7. Фрагмент кода, сгенерированного экспертной системой

```
PROGRAM Interlock1
  VAR
    level_value_026: REAL;
    valve_value_080_1: BOOL;
  END_VAR
  VAR RETAIN
    maximum_level_value : REAL;
  END_VAR

  IF level_value > maximum_level_value THEN
    valve_value_080_1 := True;
  ELSE
    valve_value_080_1 := False;
  END_IF

END_PROGRAM
```

В данном коде показан программный модуль типа «программа». Переменная `level_value_026` соответствует значению уровня воды в резервуаре, `valve_value_080_1` – состоянию клапана. Они определены с помощью конструкции `VAR ... END_VAR`. Переменная `maximum_level_value`, соответствующая уставке и определяющая максимально допустимое значение в резервуаре, должна сохраняться в энергонезависимой памяти ПЛК. Для этой цели, согласно стандарту IEC 61131-3, в конструкцию `VAR ... END_VAR` добавляется спецификатор `RETAIN` в области определения переменных. После определения переменных следует конструкция `IF ... THEN ... ELSE`, реализующая логику работы блокировки. Приведённый выше код легко встраивается в любой существующий проект, созданный в среде Veremiz, либо напрямую в механизм компиляции проекта, либо в формате XML в соответствии со стандартом PLCOpen TC6 XML.

### Заключение

В статье был продемонстрирован подход, основанный на анализе P&ID диаграмм. Реализована соответствующая экспертная система на базе CLIPS с возможностью генерации управляющего кода для ПЛК на языках стандарта IEC 61131-3. Результирующий код хорошо интегрируется с уже ранее введёнными в эксплуатацию

программными средствами для разработки управляющих алгоритмов. Такой подход применим и для генерации других программных компонентов системы управления.

В результате проведённого исследования доказана возможность реализации частичной автоматизации создания повторных, рутинных и подверженных ошибкам инженерных задач в рамках создания системы управления на основе анализа требований к ней. Также инженер-технолог может использовать реализованную систему для проверки самого себя в процессе выполнения работы над автоматизацией конкретного технологического объекта, сравнив выдачу экспертной системы со своей реализацией.

В дальнейшем планируется увеличение возможностей данной экспертной системы путем расширения вариантов анализа различных инженерных артефактов в рамках разработки программных компонент для управляющих систем в промышленной автоматизации.

### **Литература**

1. Баранов И.А., Глухов А.В. Языки стандарта IEC-61131 для вычислительных комплексов на базе отечественных микропроцессоров с архитектурой SPARC // Вопросы радиоэлектроники. – 2012. – Сер. ЭВТ. – Вып. 3.
2. Steinegger M., Zoitl A. Automated code generation for programmable logic controllers based on knowledge acquisition from engineering artifacts: Concept and case study. ETFA 2012: 1-8.
3. Drath R., Fay A. Computer-aided design and implementation of interlock control code. / Conference on Computer Aided Control Systems Design. Munich, Germany, October 4-6, 2006.
4. Specification for Representation of process control engineering requests in P&I Diagrams and for data exchange between P&ID tools and PCE-CAE tools, IEC, Technical Committee No. 3: Information structures, documentation and graphical symbols.
5. Joseph C. Giarratano. CLIPS User's Guide [Электронный ресурс]. – Режим доступа: <http://clipsrules.sourceforge.net/documentation/v630/ug.pdf>. Дата обращения 14.12.2015.

6. M. Negnevitsky. Artificial Intelligence. A Guide to Intelligent Systems, 2nd, 2005, Addison Wesley.

7. Руководство по использованию библиотеки PyCIPS [Электронный ресурс]. – Режим доступа: <http://pyclips.sourceforge.net/manual>. Дата обращения 14.12.2015.