Мезенцев К.Н.**Моделирование цифровых схем управления в программе AnyLogic***Московский Автомобильно-дорожный Государственный технический
университет МАДИ
(Россия, Москва)*

doi: 10.18411/sr-10-06-2018-03

idsp: 000001:sr-10-06-2018-03

Аннотация

В статье рассматриваются проблемы моделирования работы цифровых схем управления, построенных на базе элементарных логических операций. В качестве примера приводится схема управления двумя электродвигателями, построенная на основе RS триггера. Дается описание алгоритма разработки агентов логических операций в программе AnyLogic.

Ключевые слова: агент, интерфейс, триггер, операция, элемент, логический элемент

Abstract

The article deals with the problems of modeling the operation of digital control circuits based on elementary logical operations. As an example, a control scheme for two electric motors is presented, based on the RS flip-flop. A description of the algorithm for the development of agents of logical operations in the program AnyLogic is given.

Keywords: agent, interface, trigger, operation, element, logical element

Современные автоматические цифровые системы управления находят широкое применение различных машинах и механизмах. Для моделирования работы таких систем в целях их проектирования и исследования используются специальные программные продукты. Такие программы требуют значительного времени для их освоения, так как обладают достаточно сложным интерфейсом и набором специальных команд. Кроме того большинство таких программ распространяются на платной основе.

Программа AnyLogic обладает интуитивно понятным интерфейсом и позволяет разработчику модели на основе «элементарных» компонентов создавать с минимальными временными затратами свои собственные компоненты, в том числе и компоненты цифровых автоматических систем управления [1]. Кроме того имеется свободно распространяемая версия программы P(ersonal) L(earning) E(dition).

Цифровая схема включения двигателей

Рассмотрим схему, которая используется для подачи сигналов управляющих включением двух электродвигателей. Такая схема может быть использована для включения двигателей вентиляционной установки жилого помещения, либо складского помещения. Схема управления двигателями показана на рисунке 1. Двигатели имеют условное обозначение как «Правый» и «Левый». Схема строится на основе использования двух RS триггеров.

- Линия «Move Right» - подача сигнала для включения правого двигателя.
- Линия «Move Left» - подача сигнала для включения левого двигателя.
- Линия «Stop» - выключение работающего двигателя.

Выходные линии схемы «Right», «Left» служат для подачи сигнала запуска двигателя (таблица 1). При запуске системы управления в работу, оба двигателя оказываются включенными.

Таблица 1.

Управление двигателями

MoveRight	MoveLeft	Stop	Right	Left
0	0	1	0	0
1	0	0	1	0
1	0	1	0	0
0	1	0	0	1
0	1	1	0	0

Рисунок. 1. Схема управления двигателями

Логическая схема RS триггера в условных обозначениях US ANSI 911984 показана рисунке 2.

Рисунок. 2. схема RS триггера

Модель RS триггера

Модель состоит из двух активных классов – агентов:

ElemOR

ElemNOT

Все агенты создаются на основе «элементарного» компонента AnyLogic «Порт». Для посылки сообщений в порты записывается код на языке Java.

На рисунке 3 показан агент триггер ElemTrigger. Интерфейс агента триггера представляет собой «значок» AnyLogic с четырьмя портами, настроенными на прием и передачу сообщений – сигналов целого типа.

Рисунок 3. Агент RS триггер

Агент триггер – модель в которой размещены экземпляры агентов ElemOR и ElemNOT, эти экземпляры соединены между собой в соответствии со схемой 2. Поэтому сначала требуется создать два вспомогательных агента, реализующих операцию OR – «ИЛИ» и NOT – «НЕ». Агенты этих элементов должны обладать тремя портами [2,3], настроенными для подачи и приема сигнала целого типа и значком в соответствии с рисунком 3.

Агент ElemNOT содержит глобальную переменную vX логического типа. Значение по умолчанию false. Левый порт содержит код, выполняемый при получении сообщения - сигнала:

```
if (msg.intValue()==1) {vX=true;}
else {vX=false;}
```

Для преобразования сигнала служит динамическая переменная [2,3] flowOutSignal. Переменная осуществляет вызов функции getOutSignal(), она возвращает значение целого типа. Код функции:

```
if (vX) {outY.send(new Integer("0"));}
else
{outY.send(new Integer("1"));}
return 0;
```

Здесь outY идентификатор левого порта.

Агент ElemOR должен обладать глобальными переменными vX1 и vX2 логического типа. Значение по умолчанию false. Для входных портов нужно ввести код при получении сообщения в виде:

```
if (msg.intValue()==1){vX1=true;}
else {vX1=false;}
if (msg.intValue()==1){vX2=true;}
else {vX2=false;}
```

Элемент содержит динамическую переменную `flowOutSignal`, которая служит для вызова функции `getSignal()` получения выходного сигнала элемента. Функция возвращает значение целого типа.

```
boolean res = vX1 || vX2;
if (res) {outY.send(new Integer("1"));}
else {outY.send(new Integer("0"));}
return 0;
```

Где `outY` идентификатор правого порта.

Модель цифровой схемы включения двигателей

Агент для тестирования работы схемы показан на рисунке 4. Агент содержит два экземпляра агента `ElemTrigger`. Для подачи сигналов используются экземпляры агента `ToggleSignal`, а для отображения сигналов на выходных линиях служат экземпляры агента `DispSignal`. Агенты коммутированы в соответствии со схемой рисунка 1.

Агент `ToggleSignal` содержит кнопку, при нажатии которой на выходном правом порте `outY` выдается сигнал равный 1 или 0. Значение этого сигнала отображается рядом с портом. Код кнопки:

```
if (vFlag) {
vFlag=false;
text.setText("1");
outY.send(new Integer("1")); }
else {
vFlag=true;
text.setText("0");
outY.send(new Integer("0")); }
```

Где `vFlag` – переменная логического типа элемента, значение по умолчанию равно «Истина». Для генерации по умолчанию на выходе порта значения равного нулю нужно использовать простое событие `event [2,3]` выполняющееся один раз с кодом:

```
outY.send(new Integer("0"));
```

Агент `ElemDisp` обладает левым входным портом `inX` для приема - отправки сигнала целого типа. При получении сигнала порт должен отображать полученный сигнал как надпись, на элементе используя код:

```
text.setText(">" + msg.intValue());
```

Рисунок. 4. Тестирование схемы управления, начальное состояние

Рисунок. 5. Тестирование схемы управления, левый двигатель остановлен

Начальное состояние схемы – оба двигателя включены (рисунок 4). На рисунке 5 показано состояние включения правого двигателя.

Заключение

Программа AnyLogic позволяет создавать визуальные компоненты элементарных логических операций с коммутационным интерфейсом. Эти компоненты используются как «строительные» элементы для создания схем управления различного уровня сложности.

Для создания компонентов – элементов цифровых схем управления требуется один «элементарный» элемент «Порт» AnyLogic. Функциональные особенности порта определяются кодом Java, причем объем этого кода минимальный.

1. Мезенцев К.Н. Моделирование в примерах и задачах в среде AnyLogic /К.Н. Мезенцев. Lap LAMBERT Academic Publishing, 2013 – 205 с.
2. Мезенцев, К.Н. Моделирование систем в среде AnyLogic 6.4.1. Учебное пособие. Часть 1. /Под редакцией Заслуженного деятеля науки РФ, д.т.н., профессора А.Б.Николаева/К.Н. Мезенцев // МАДИ (ГТУ), 2011. -105с.
3. Мезенцев, К.Н. Учебное пособие Моделирование систем в среде AnyLogic 6.4.1. Часть 2 /Под редакцией Заслуженного деятеля науки РФ, д.т.н., профессора А.Б.Николаева/К.Н. Мезенцев // МАДИ (ГТУ), 2011. - 112с.

Осипов В.И.

Исследование влияния соглашения об уровне качества на выполнение заявок пользователей на примере трех разных систем Service Desk

ФГБОУ ВО "Национальный исследовательский университет "МЭИ"
(Россия, Москва)

doi: 10.18411/sr-10-06-2018-04

idsp: 000001:sr-10-06-2018-04

Аннотация

В статье рассматривается полезность соглашения об уровне качества (далее SLA). Производится обзор трех систем регистрации заявок на предприятии, а также выполняется анализ SLA для каждой из трех систем. В результате анализа выясняется, что SLA с наименьшим количеством посредников положительно сказывается на скорости работы отдела технической поддержки предприятия.

Ключевые слова: SD, Service Desk, SLA, регистрация заявок, соглашение об уровне качества.