

Реализация конечных автоматов в масштабных проектах автоматизации

И.М. Сафаров, Д.И. Хаматханов

Казанский государственный энергетический университет

Аннотация: Несмотря на стремительную поступь технологий, надежные методы программирования, в частности в технологиях автоматизации, развиваются крайне медленно. Поэтому эффективное программирование, особенно в масштабных проектах автоматизации, является задачей сегодняшнего дня. В данной научно-исследовательской работе рассматривается шаблон состояния как метод проектирования конечных автоматов на примере торгового автомата.

Ключевые слова: Конечный автомат, Structured text, МЭК 61131-3, CoDeSys, объектно-ориентированное программирование, CASE, наследование, интерфейсы, методы, состояние.

Конечные автоматы используются повсеместно, особенно в технологиях автоматизации. Типичный пример реализации автоматов это оператор CASE. Некоторые программисты в местах перехода состояний используют методы. Очевидно, что более крупные приложения не могут быть реализованы таким образом. Ясность полностью теряется после нескольких переходов состояний. Это говорит о том, что данные методы не масштабируются. Более того, каждый раз, когда добавляется новое состояние, необходимо править несколько методов или участки кода, где используется оператор CASE. Рассмотренная далее концепция позволяет с легкостью структурировать процессы автоматизации, что обеспечивает удобство в проектировании системы и дальнейшей ее масштабируемости [1].

Рассмотрим применение шаблона на примере торгового автомата, который выдает продукт после вставки монеты и нажатия кнопки. Условимся, что при отсутствии товара, монета возвращается.

Конечный автомат удобно представить в виде диаграммы состояний UML:

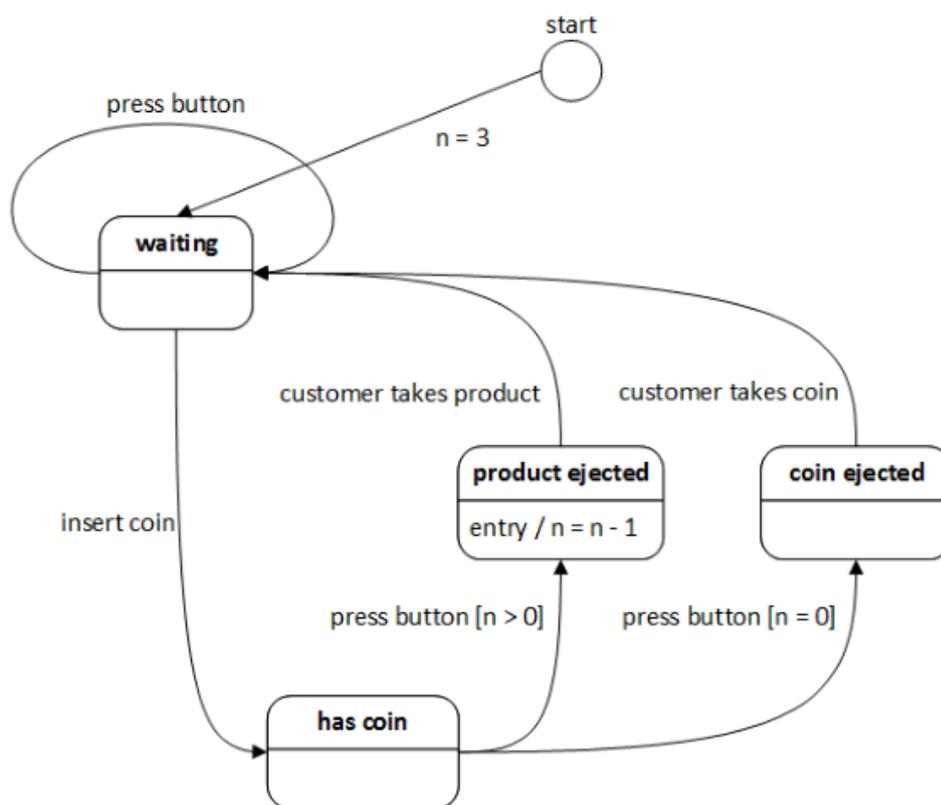


Рис. 1. – UML-диаграмма состояний, описывающая торговый автомат

Логика торгового автомата поместим в функциональный блок (ФБ) FB_Machine, где входные переменные блока это события, а на выходные переменные запишем количество доступных продуктов и текущее состояние.

Листинг 1. – Функциональный блок FB_Machine

```

1: FUNCTION_BLOCK PUBLIC FB_Machine
2: VAR_INPUT
3:   bButton : BOOL;
4:   bInsertCoin : BOOL;
5:   bTakeProduct : BOOL;
6:   bTakeCoin : BOOL;
7: END_VAR
8: VAR_OUTPUT
9:   eState : E_States;
10:  nProducts : UINT;
11: END_VAR
    
```

Листинг 2. – Собственный тип E_States

```

1: TYPE E_States :
2: (
3:   eWaiting := 0,
4:   eHasCoin,
5:   eProductEjected,
    
```

```
6:     eCoinEjected  
7: );  
8: END_TYPE
```

Шаблон разбивает все возможные состояния торгового автомата на несколько объектов. Таким образом, каждый возможный статус представлен отдельным ФБ. Эти статусные ФБ содержат все поведения для соответствующего состояния. Благодаря этому, новый статус может быть введен без изменения исходного кода исходных блоков.

Каждое действие (CustomerTakesCoin(), CustomerTakesProduct(), InsertCoin() и PressButton()) может выполняться в любом состоянии. Таким образом, все статусные ФБ используют один и тот же интерфейс:

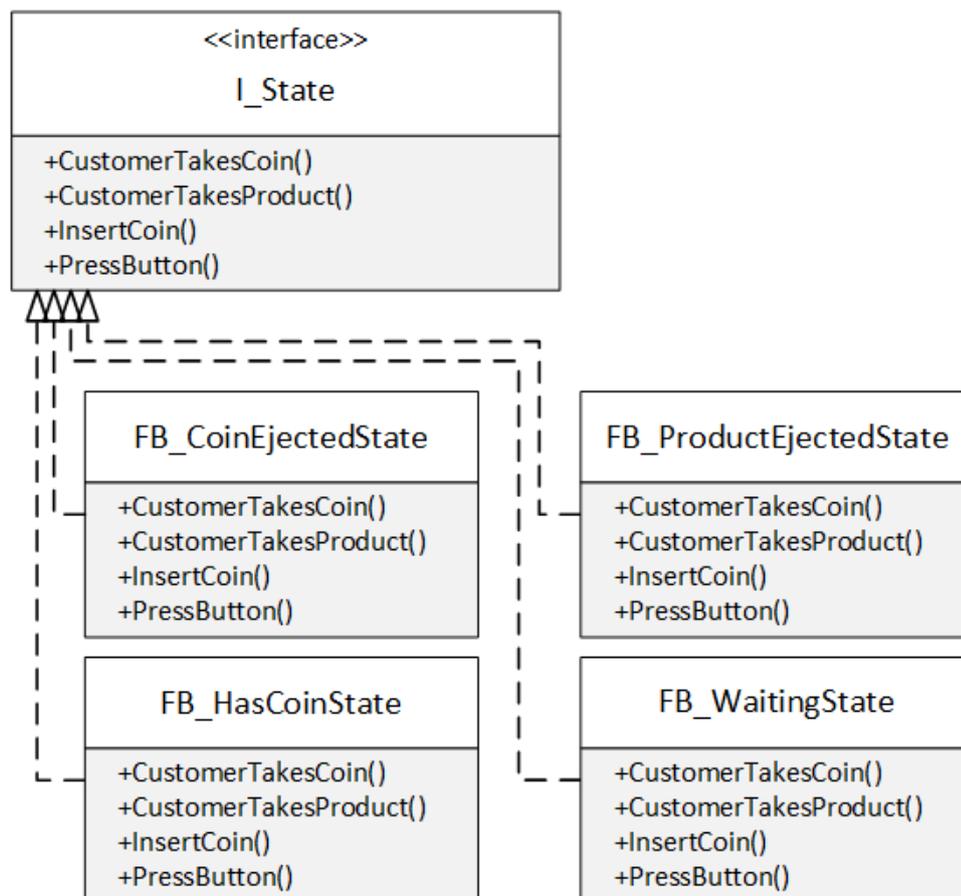


Рис. 3. – UML-диаграмма компонентов торгового автомата

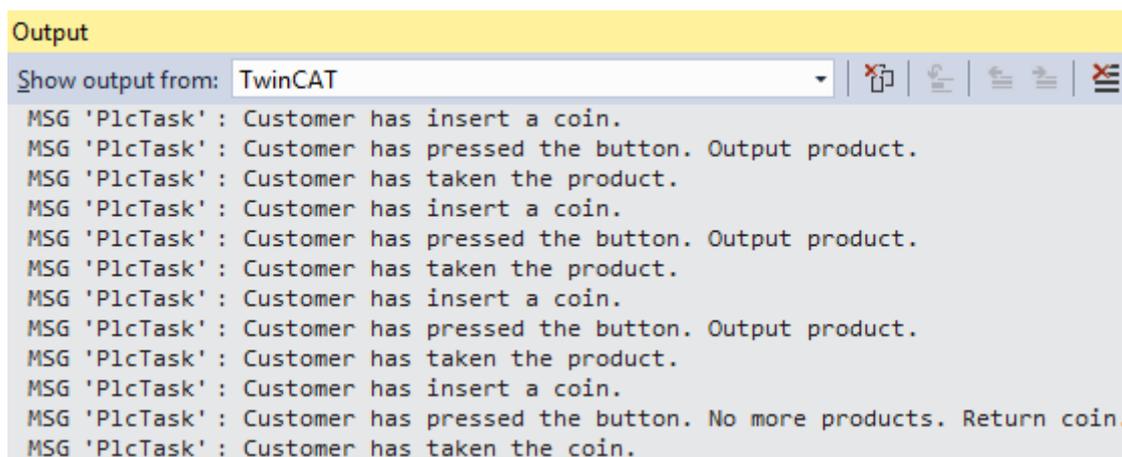
FB_Machine объединяет этот интерфейс (строка 9), который делегирует вызов метода соответствующим ФБ состояния (строки 30, 34, 38 и 42) [9].

Листинг 7. – Реализация интерфейса

```
9: FUNCTION_BLOCK PUBLIC FB_Machine
10: VAR_INPUT
11:   bButton      : BOOL;
12:   bInsertCoin  : BOOL;
13:   bTakeProduct : BOOL;
14:   bTakeCoin    : BOOL;
15: END_VAR
16: VAR_OUTPUT
17:   ipState      : I_State := fbWaitingState;
18:   nProducts    : UINT;
19: END_VAR
20: VAR
21:   fbCoinEjectedState : FB_CoinEjectedState(THIS);
22:   fbHasCoinState     : FB_HasCoinState(THIS);
23:   fbProductEjectedState : FB_ProductEjectedState(THIS);
24:   fbWaitingState     : FB_WaitingState(THIS);
25:   rtrigButton        : R_TRIG;
26:   rtrigInsertCoin    : R_TRIG;
27:   rtrigTakeProduct   : R_TRIG;
28:   rtrigTakeCoin      : R_TRIG;
29: END_VAR
30: rtrigButton(CLK := bButton);
31: rtrigInsertCoin(CLK := bInsertCoin);
32: rtrigTakeProduct(CLK := bTakeProduct);
33: rtrigTakeCoin(CLK := bTakeCoin);
34: IF (rtrigButton.Q) THEN
35:   ipState.PressButton();
36: END_IF
37: IF (rtrigInsertCoin.Q) THEN
38:   ipState.InsertCoin();
39: END_IF
40: IF (rtrigTakeProduct.Q) THEN
41:   ipState.CustomerTakesProduct();
42: END_IF
43: IF (rtrigTakeCoin.Q) THEN
44:   ipState.CustomerTakesCoin();
45: END_IF
```

Листинг 10. – Проверка функционального блока FB_Machine

```
46: PROGRAM MAIN
47: VAR
48:   fbMachine : FB_Machine(3);
49:   sState    : STRING;
50:   bButton   : BOOL;
51:   bInsertCoin : BOOL;
52:   bTakeProduct : BOOL;
53:   bTakeCoin  : BOOL;
54: END_VAR
55:
56: fbMachine(bButton := bButton,
57:           bInsertCoin := bInsertCoin,
58:           bTakeProduct := bTakeProduct,
59:           bTakeCoin := bTakeCoin);
60: sState := fbMachine.ipState.Description;
61:
62: bButton := FALSE;
63: bInsertCoin := FALSE;
64: bTakeProduct := FALSE;
65: bTakeCoin := FALSE;
```



```
Output
Show output from: TwinCAT
MSG 'PlcTask': Customer has insert a coin.
MSG 'PlcTask': Customer has pressed the button. Output product.
MSG 'PlcTask': Customer has taken the product.
MSG 'PlcTask': Customer has insert a coin.
MSG 'PlcTask': Customer has pressed the button. Output product.
MSG 'PlcTask': Customer has taken the product.
MSG 'PlcTask': Customer has insert a coin.
MSG 'PlcTask': Customer has pressed the button. Output product.
MSG 'PlcTask': Customer has taken the product.
MSG 'PlcTask': Customer has insert a coin.
MSG 'PlcTask': Customer has pressed the button. No more products. Return coin.
MSG 'PlcTask': Customer has taken the coin.
```

Рис. 5. – Результат программы в системном журнале TwinCAT

Распределение обязанностей каждого ФБ делает этот подход очень гибким и гораздо более надежным для расширений. Это особенно заметно, когда статус ФБ становится более обширным, где конечный автомат может управлять сложным процессом, в котором каждый статус ФБ содержит дополнительные подпроцессы.

Литература

1. Новиков Д.А. Теория управления организационными системами. 2-е изд. М.: Физматлит, 2007. 604 с.
2. Сафаров И.М., Хаматханов Д.И. Автоматизированная система управления параметрами теплоносителя с удаленным доступом // Инженерный вестник Дона, 2018, №2 URL: ivdon.ru/ru/magazine/archive/N2y2018/4912.
3. Шамгунов Н.Н., Корнеев Г.А. State Machine — новый паттерн объектно-ориентированного проектирования // Информационно-управляющие системы, 2004, №5 с. 13-25.
4. Белоусов А.И., Ткачев С.Б. Дискретная математика. М.: МГТУ, 2006. С. 460-587.
5. Руководство пользователя по программированию ПЛК в CoDeSys 2.3. Смоленск: ПК Пролог, 2006. 453 с.

6. Капустин Н.М. Автоматизация производственных процессов в машиностроении: Учеб. для вузов / Под ред. Н.М. Капустина. М.: Высшая школа, 2004. 415 с.

7. МВА8 Модуль ввода аналоговый измерительный: Руководство по эксплуатации. М.: АТЛАС-ПРЕСС, 2008. 90 с.

8. Вадутов О. Настройка типовых регуляторов по методу Циглера-Никольса. Методические указания к выполнению лабораторной работы. Томск. 2014. 10 с.

9. Софиева Ю.Н., Абрамов К.В. Применение пакета моделирующих программ ChemCAD в учебно- тренировочных комплексах для изучения систем автоматизации ректификационных установок // Инженерный вестник Дона, 2012, №1 URL: ivdon.ru/ru/magazine/archive/n1y2012/619.

10. Матвеева Л.Л. Информационные технологии в менеджменте. М., 2010. 187 с.

11. McGraw-Hill, Sybil P. Parker Dictionary of Scientific and Technical Terms. 6th Edition. New York: Merck, 2002. 2380 p.

12. Grigoryuk E.N ., Bulkin V.V. Problems of Automation and Management Principles. Information Flow in Manufacturing // IOP Conference Series: Materials Science and Engineering, 2017, Volume 221, conference 1 URL: doi.org/10.1088/1755-1315/221/1/012006.

References

1. Novikov D. A. Teoriya upravleniya organizatsionnymi sistemami [Organization Management Theory]. vol 2. М.: Fizmatlit, 2007. 604 p.

2. Safarov I.M., Khamatkhanov D.I. Inženernyj vestnik Dona (Rus), 2018, №2 URL: ivdon.ru/ru/magazine/archive/N2y2018/4912.

3. Shamgunov N.N., Korneyev G.A. Informatsionno-upravlyayushchiye sistemy, 2004, №5 pp. 13-25.

4. Belousov A.I., Tkachev S.B. Diskretnaya matematika [Discrete Math]. M.: MGTU, 2006. pp. 460-587.
5. Rukovodstvo polzovatelya po programmirovaniyu PLK v CoDeSys 2.3 [PLC Programming User Guide for CodeSys 2.3]. Smolensk: PK Prolog, 2006. 453 p.
6. Kapustin N.M. Avtomatizatsiya proizvodstvennykh protsessov v mashinostroenii: Ucheb. Dlya Vuzov [Automation of production processes in mechanical engineering: Proc. For universities]. Pod. Red. N.M. Kapustina. M.: Vysshaya shkola, 2004. 415 p.
7. MBA8 Modul' vvoda analogoviy izmeritelnyy: Rukovodstvo po ekspluatatsii [MVA8 Analog Measuring Input Module: Operation Manual]. M.: Atlas-Press, 2008. 90 p.
8. Vadutov O. Nastroyka tipovykh regulyatorov po metodu Tsiglera-Nikolsa. Metodicheskiye ukazaniya k vypolneniyu laboratornoy raboty [Setting type regulators according to the Ziegler-Nichols method. Guidelines for the implementation of laboratory work]. Tomsk. 2014. 10 p.
9. Sofieva Iu.N., Abramov K.V. Inzhenernyy vestnik Dona (Rus), 2012, №1 URL: ivdon.ru/ru/magazine/archive/n1y2012/619.
10. Matveeva L. L. Informatsionnye tehnologii v menedjmente [Management Information Technology]. M., 2010. 187 p.
11. McGraw-Hill, Sybil P. Parker Dictionary of Scientific and Technical Terms. 6th Edition. New York: Merck, 2002. 2380 p.
12. Grigoryuk E.N., Bulkin V.V. IOP Conference Series: Materials Science and Engineering, 2017, Volume 221, conference 1 URL: doi.org/10.1088/1755-1315/221/1/012006.