

Опубликовано в материалах конференции “Microsoft Research Academic Days in St. Petersburg, April 21-23, 2004”

Язык автоматного программирования с компиляцией в *Microsoft CLR*

Н.Н.Шамгунов, А.А.Шалыто (СПбГУ ИТМО Тел.: (812) 247-95-45, email: shalyto@mail.ifmo.ru)

В настоящее время активно развивается технология автоматного программирования [Шалыто]. Совершенствование указанной технологии связано с совместным использованием автоматов и классов. Этот вопрос исследуется в проектах, опубликованных на сайте <http://is.ifmo.ru>, во многих из которых используется конструкция *switch*, которая не является «объектной».

Цель настоящей работы состоит в разработке нового объектно-ориентированного языка программирования, который позволяет эффективно работать с конечными автоматами на основе объектно-ориентированной модели.

Наиболее близким подходом для решения указанной задачи является применение любого объектно-ориентированного языка и реализация на нем паттерна *State* [Гамма]. Такой подход неэффективен, так как для каждого автомата необходимо ввести столько классов, сколько состояний в реализуемом автомате плюс один (базовый) и для доступа к данным автомата из объектов, реализующих состояния, необходимо обращаться к контексту [Гамма], хранящему эти данные, что усложняет код.

Авторы считают, что одним из важнейших факторов, ограничивающих широкое использование автоматов в разработке программного обеспечения, является отсутствие непосредственной их поддержки в современных языках программирования. Для устранения этого недостатка разработан новый язык программирования *State*. Язык построен на базе языка *C#*, в который введены синтаксические конструкции «*automaton*» и «*state*». Программа на этом языке компилируется в сборку *Microsoft CLR* [Richter].

Automaton (далее автомат) – это расширение понятия *class* за счет использования понятия «состояние». В теле автомата можно определить одно или несколько состояний *state*, внутри каждого, из которых описывается набор методов. Это позволяет реализовывать паттерн *State* не в виде совокупности классов, а внутри одной логической конструкции. Автомат может наследоваться как от класса, так и от другого автомата.

Для компиляции языка разработан препроцессор, преобразующий код на языке *State* в код на языке *C#*. Препроцессор преобразует автомат в класс, который может быть скомпилирован в *Microsoft CLR*. Таким образом, обеспечена возможность работы с автоматом как с единым объектом, как на этапах проектирования и кодирования, так и на этапе использования.

В качестве примера реализуем простейший автомат с двумя состояниями. При переходе из первого состояния во второе при входном воздействии *move* формируется выходное воздействие (на консоль выводится строка *State1*). На петле во втором состоянии при том же входном воздействии на консоль выводится строка *State2*. Этот автомат на языке *State* описывается следующим образом.

```
automaton Sample
{
    enum State
    {
        state1,
        state2,
    }
    State _state = State.state1;
```

```

state ( _state == State.state1 )
{
    public void move ()
    {
        Console.WriteLine ( "state1" );
        _state = State.state2;
    }
}
state ( _state == State.state2 )
{
    public void move ()
    {
        Console.WriteLine ( "state2" );
        _state = State.state2;
    }
}
}

```

Применение автоматов в программировании в настоящее время активно развивается. Так, в частности, в *Microsoft Research* разработан язык *ASML* (Abstract State Machine Language) [Гуревич]. Предлагаемый язык рассчитан на более широкую аудиторию, так как является расширением языка *C#*.