

ВЫСОКОУРОВНЕВАЯ ФОРМА СИНТЕЗА СИСТЕМ УПРАВЛЕНИЯ

В.С. Выхованец

*Институт проблем управления им В. А. Трапезникова РАН, г. Москва, ул. Профсоюзная, 65
E-mail. vykhovanets@mail.ru*

С практической точки зрения успех проектирования системы управления зависит от используемой; методологии программирования - интегрального, сформулированного в виде методов, методик и приемов, подхода к основным стадиям проектирования программных средств. Весьма перспективной представляется объектно-ориентированная (ОО) методология. На практике дела с ОО подходом обстоят не так хорошо. Как отмечено в [1] ОО модель далека от реальности. Корни проблемы – в объективной трудоемкости, интеллектуальной и технологической сложности процесса программирования.

В развитие ОО методологии для построения систем управления с плохо формализуемой предметной областью предлагается использование контекстных языковых средств [2], в основу которых положена продукционная модель знаний. Контекстный язык близок к естественным языкам. Основными синтаксическими единицами языка является определение некоторого понятия посредством его описания в виде совокупности предложений, разъясняющих содержание этого понятия. Понятия образуют иерархическую структуру, соответствующую иерархии классов при ОО программировании. При определении предложений определяются и действия – код, необходимый для правильной интерпретации описываемого понятия в контексте каждого из предложений. Форма представления кода – предложения, содержащие ранее описанные понятия. Первичные понятия (числа, строки, и т.д.) привязывается к аппаратной платформе. Терминальные константы заключаются в одинарные кавычки. Для описания терминальных конструкций используется язык регулярных выражений (в двойных кавычках). В квадратных скобках задается интерпретация предложения во время компиляции, а в фигурных скобках – во время исполнения программы. Единственным неопределяемым термом языка является знак '@', осуществляющий запись некоторой сущности в область кода. Для ссылок на понятия и термы, образующие структуру предложения, используется конструкция вида '%n', где n - номер слова в предложении. Пример программы:

```
essence is essence:
    essence '@'.
literal is essence:
    "'*.'" [%0 @], literal.
number is essence, uses literal:
    "-?[0-9]+" [%0 @], number;
    '(' number ')', number;
    '-' number ['neg' @], number;
    number '*' number ['mul' @], number;
    number '/' number ['div' @], number;
    number '+' number ['add' @], number;
    number '-' number ['sub' @], number.
boolean is number:
    'false' [0 @], boolean;
    'true' [-1 @], boolean;
    '(' boolean ')', boolean;
    'not' boolean ['not' @], boolean;
    boolean 'and' boolean ['and' @], boolean;
    boolean 'or' boolean ['or' @], boolean;
    boolean 'imply' boolean, boolean { not %0 or %1 }.
```

Литература

1. Lewis T. Software Architectures: Divine plan or digital Darwinism // Computer. № 8. 1996. P. 13-15.
2. Выхованец В. С., Иосенкин С. Я. Применение технологии контекстного программирования для решения больших прикладных задач // Труды международной конференции «Параллельные вычисления и задачи управления» (РАСО'2001). – М., 2001. – С.(4)-121-139.