

УДК 519.681

## КОНТЕКСТНОЕ ПРОГРАММИРОВАНИЕ В МОДЕЛИРОВАНИИ

*Иосенкин В.Я., Выхованец В.С.*

Решение любой задачи всегда связано с проникновением в соответствующую предметную область. Предметы и события, лежащие в основе понимания информации, необходимой для решения задачи, составляют предметную область задачи. Предметная область представляется состоящей из абстрактных объектов (сущностей). Сущности предметной области находятся в определенных отношениях друг к другу (ассоциациях), которые также можно рассматривать как сущности и включать в предметную область, Совокупность сущностей, между которыми наблюдаются различные отношения подобия, составляет группу сущностей, являющуюся новой сущностью этой предметной области.

Отношения между сущностями выражаются с помощью суждений (мысленно возможных ситуаций, которые могут иметь или не иметь места для рассматриваемых сущностей). В языке (формальном или естественном) суждениям соответствуют лексические конструкции (предложения) Предложения также можно рассматривать как, сущности и включать в предметную область. Совокупность лексических конструкций, характеризующих предметную область, будем называть знаниями об этой предметной области.

Универсальным языком представления знаний является естественный язык. Использование естественного языка в компьютерных системах наталкивается на трудности из-за присущих ему нерегулярностей, двусмысленностей, неоднозначностей, неопределенностей и т.п. Главное препятствие заключается в отсутствии формальной семантики естественного языка, которая имела бы достаточно эффективную операционную поддержку. Как известно, современные технологии программирования реализуются в рамках языков программирования, порождаемых узким классом контекстно-свободных формальных грамматик Ограничением выразительных качеств этих языков программирования является бесконтекстность правил вывода, что не согласуется с явлением контекстной интерпретации слов в естественных языках и, как следствие этого, уменьшает надежность программирования, устанавливает непреодолимый барьер между языком программирования и естественным языком.

В рамках компонентой технологии предлагается контекстная система программирования, позволяющая снизить расхода на программирование, повысить надежность и безопасность кода [1]. Контекстный язык близок к естественным языкам и представлен в виде полуконтекстной формальной грамматики. В качестве вычислительного механизма использован магазинный автомат, имеющий 2 (3) стека: стек операндов, стек адресов возврата (стек контекста) Использование третьего стека позволяет реализовать динамическое связывание при произвольном контексте и произвести динамическую генерацию кода. Для привязки программы к конкретной вычислительной системе некоторые понятия, называемые первичными, определяются низкоуровневыми средствами, например, на языке ассемблера.

Приведем грамматику языка контекстного программирования Esse [2]. Язык Esse позволяет приблизить процесс проектирования программных средств к парадигмам постановки и решения многих прикладных задач.

```
program ::= '{' knowledge}' text
knowledge ::= essence | essence '.' knowledge
essence ::= name 'is' visible nation 'that' description
visible ::= 'public' | 'protected' | 'private'
description ::= sentence | sentence ';' description
sentence ::= context words 'makes' nations 'so' definition
```

**context** ::= '[' nations '  
**words** ::= **term** | **term words** | **nation local** | **nation local words**  
**nations** ::= **nation local** | **nation local nations**  
**definition** ::= **token** | **token definition**  
**local** ::= 'as' **name**  
**term** ::= "' token "' | "' token "' **term**

Программа состоит из декларации знаний о предметной области **knowledge** и описания некоторой ситуации **text**, которая является исполняемой частью программы. Определяемые сущности **essence** образуют понятия **nation** с именами **name** находящиеся в отношении наследования. Каждое понятие определяется лексическими конструкциями **sentence**. В последних задается контекст их применения **context**, последовательность **words** терминальных знаков **term** и понятий **nation**, производящих новый контекст **result** путем выполнения действий **definition**.

В **definition** используются ранее описанные лексические конструкции и локальные значения понятий **local** из определяемой конструкции, посредством чего формируются значения нового контекста.

Лексемы **token** представляют собой последовательности знаков, разделенных пробельными знаками или знаками пунктуации, которые, в свою очередь, также являются лексемами.

Для привязки программы к конкретной вычислительной системе некоторые понятия, называемые первичными, определяются низкоуровневыми средствами **assembler** в виде последовательности команд **command**.

Предлагаемый язык наиболее близок к языку программирования FORTH и является его полной объектно-ориентированной модификацией без использования обратной польской записи выражений.

Выразительные возможности языка Esse соответствуют полуконтекстной формальной грамматике, в которой терминальные и нетерминальные знаки **B...B** интерпретируются в зависимости от левого контекста **A...A** и в результате интерпретации получается новый контекст **C...C**:

$$A \dots A B \dots B \rightarrow C \dots C$$

Код представляется в виде дерева сущностей **essence**, состоящих из синтаксических конструкций **sentence**, определяемых через ранее декларированные сущности или в командах **command** некоторой виртуальной машины. Дерево мигрирующего кода подгружается в заданную точку дерева сущностей виртуальной машины.

Программа состоит из знаний, характеризующих предметную область (декларативные знания) и описания некоторой ситуации, которая является исполняемой частью программы (императивная часть). Императивная часть побуждает к определенным действиям, заложенным в словаре для данного действия, но с учетом ситуации созданной в императивной части (выполнение осуществляется с учетом контекста). Т.е. такая декомпозиция позволяет ожидать разных действий, как в результате изменения исполняемой части, так и описательной, и, кроме того, в зависимости от подключенного в данный момент дерева словарей.

Перечень терминальных объектов, которые могут обрабатываться, задается как при проектировании системы (базовые сущности), так и в коде самой программы (пользовательские сущности). Ими могут быть целые вещественные числа, идентификаторы, строки, списки и т. п. Семантика терминальных объектов определяется набором команд, оперирующих с ними, например: арифметические действия над числами, сравнение между собой строк или идентификаторов, операции ввода-вывода, включающие необходимые трансформации представлений, и т. д.

Использование языка контекстного программирования позволяет сократить число ошибок, возникающих из-за семантического разрыва между применяемыми языковыми

Моделирование. Теория, методы и средства: Материалы междунар. науч.-практ. конф. – Новочеркасск: УПЦ «Набла» ЮРГТУ(НПИ), 2001. – Ч. 7. – С. 49-51.

средствами и математической моделью предметной области и, тем самым, повысить скорость и надежность разработки программ, в особенности связанных с пониманием естественно-языковых текстов, планированием поведения, обучения, принятия решений, управления изменениями среды и др.

#### Литература

1. Выхованец В.С. Контекстная технология программирования // Телеком-99: Труды IV Междунар. науч.-техн. конф по телекоммуникациям. – Одесса, 1999. – С. 116-119.

2. Выхованец В. С. Технология безопасного программирования // Проблемы управления безопасностью сложных систем: Тезисы докладов 8-й Междунар. конф. – М., 2000. – Т. 2. – С. 89-91.

*3300, Молдова, Тирасполь, ул. 25 Октября, д. 128, Приднестровский государственный университет им Т.Г. Шевченко, т (233) 3-64-22, e-mail: srb@tdsu.tirastel.md*