

## ВЫСОКОУРОВНЕВАЯ ФОРМА МИГРИРУЮЩЕГО КОДА

**В.С. Выхованец, О.Е. Дьяченко, Д.А. Купченко**

*Приднестровский государственный университет  
Тирасполь, Республика Молдова*

### Секционный доклад

Использование мигрирующего кода в глобальных компьютерных сетях порождает проблему безопасности его выполнения. Наиболее распространенной формой мигрирующего кода в сети Интернет являются Java-апплеты, состоящие из интерпретируемых команд виртуальной Java-машины. Безопасное использование апплетов обеспечивается двумя методами:

- выполнением апплета 'в специальной операционной среде (метод "песочницы");
- сертификация апплетов перед выполнением (метод "маркера доверия").

В технологии Juice мигрирующий код представляется в специфическом формате, называемом "тонким бинарным кодом". Тонкий бинарный код получается на этапе синтаксического анализа программы и представляется в виде семантического дерева, в котором сохранена информация о структуре программы, области видимости переменных и т.д.

Недостатками описанных форм представления мигрирующего кода являются:

- невозможность семантического контроля за ходом выполнения Java-апплетов из-за низкоуровневого промежуточного кода;
- получаемый код является промежуточным результатом компиляции с некоторого языка высокого уровня; это увеличивает объем мигрирующей части кода из-за необходимости трансляции программы в форму, не обходимую
  - для используемой виртуальной машины;
  - наличие семантического разрыва между языком программирования высокого уровня и математической моделью предметной области и между математической моделью предметной области и ее описанием на естественных языках.

Приведем грамматику разработанного языка контекстного программирования Esse.

```
program ::= '{' knowledge '}' text
knowledge ::= essence | essence '.' knowledge
essence ::= name 'is' visible notion 'that' description
visible ::= 'public' | 'protected' | 'private'
description ::= sentence | sentence ';' description
Sentence ::= context words 'makes' notions 'so' definition
context ::= '[' notions ']'
words ::= term | term words | nation local j nation local words
notions ::= notion local | notion local notions
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 notion 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 [1], в которой терминальные и нетерминальные знаки В...В интерпретируются в зависимости от левого контекста А...Аи в результате интерпретации получается новый контекст С...С:

$$A...A\ B...B \rightarrow A...A\ C...C.$$

Мигрирующий 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.

Использование контекстной технологии программирования и высокоуровневой формы представления мигрирующего кода позволяет сократить семантический разрыв между математической моделью и языковыми средствами, снизить число ошибок программирования, улучшает условия контроля за безопасностью кода, повышает надежность разработки программ.

Применение технологии исполнения мигрирующего кода Esse также позволит сократить интенсивность обмена в глобальной сети путем предварительного согласования (подгрузки, изменения) высокоуровневых протоколов.

Для актуализации взаимодействия в терминах (понятиях), в которых формулируется решение задачи на клиентской стороне.

### Литература:

1. Выхованец В.С. Контекстная технология программирования //Труды IV международной научно-технической конференции по телекоммуникациям (Телеком-99), 14-17 сентября 1999, Одесса, с.116-120.

## THE HIGH-LEVEL FORM OF MIGRATION CODE

V.S.Vykhovanets, O.E.Diachenko, D.A.Kupchenko

*Trans-Dniester State University  
Tiraspol, Republic of Moldova*

The section report

In a paper is considered a safety of migration code used in the global networks. The grammar of the context sensitive programming language Esse is given. The migration code is represented as a tree of essences. The essence consists of syntactic constructions.

The syntactic construction is defined through earlier declared essences or by the commands of some virtual machine. The tree of migration code is loaded at given point of the essences tree other network system.

Use of contextual technology of programming and the high-level forms of the representation of the migration code allows to reduce semantic break between mathematical model and programming language, means to lower the number of programming mistakes, improves conditions of the control behind safety code, raises reliability of the programs.