

УДК 004.432.45

Е.Г. Лутцев

## ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ, ПРИБЛИЖЕННЫХ К ЕСТЕСТВЕННОМУ: ОБЗОР ЛИТЕРАТУРЫ

*Рассматривается ряд научных статей, посвященных вопросам программирования на языках, приближенных к естественному. Дается описание классических подходов к созданию естественных языков программирования и новых подходов, которые сделали разработку таких языков практичнее. Приводится сравнение одного из языков высокого уровня – CLIPS с естественными языками программирования. Анализируется монография на русском языке, посвященная естественно-языковым интерфейсам.*

### Введение

Естественный язык (ЕЯ) – вербальный язык общения людей между собой. Попытки программирования на ЕЯ делались и несколько десятилетий назад, но современные исследователи сходятся на том, что только сейчас появились необходимые предпосылки для успешного развития этого направления.

В научной среде известен феномен, когда именитый ученый, высказываясь негативно о каком-то направлении исследований, фактически уничтожает его, прерывая поток публикаций на несколько десятилетий. Подобный эффект произвела статья Э.В. Дейкстра «О глупости программирования на естественном языке» [1]. В ней, как кажется автору, необоснованно проведена полная аналогия между программированием и математикой, хотя программирование многогранно и в значительной степени пересекается с психологией. Ранние подходы к созданию языка, напоминающего естественный, страдали и от противоположного недостатка – полного игнорирования математических абстракций, удобной нотации и сокращений. В результате были созданы переусложненные языки, например такие, как КОБОЛ, где сложение двух чисел выражается инструкцией ADD A B GIVING ANS. Современные программисты часто не используют знания математики, выходящие за пределы начальной школы.

В Интернете встречается значительное количество ресурсов со злоупотреблением термином «программирование на естественном языке», поэтому при выборе материала для чтения следует проявлять осмотрительность.

В настоящей статье рассматривается письменная речь, т. е. жесты, эмоциональный тон и прочие невербальные элементы не учитываются.

### 1. Языки программирования, использующие классические технологии обработки ЕЯ

В связи с новизной рассматриваемой темы ей пока не посвящена ни одна монография, существует только небольшое количество статей. Программы, описанные во всех рассмотренных источниках, осуществляют ввод-вывод в текстовом виде.

В настоящее время развитие систем программирования на ЕЯ происходит преимущественно по двум направлениям: создание универсальных систем, но с ограниченным словарем и онтологической моделью и создание узких языков для конкретных проблемных областей. Пример первого направления – язык программирования Inform 7 [2]. Важной его характеристикой является сильный уклон в сторону декларативного, основанного на правилах стиля программирования и способности выводить типы и свойства объектов на основе того, как они используются. Например, высказывание «Джон носит шляпу» создает «лицо» по имени Джон (так как только люди способны носить вещи), создает «вещь» со свойством «носимая» (поскольку только объекты с пометкой «носимые» можно носить) и задает Джону свойство «носит шляпу».

Еще одним существенным аспектом языка Inform 7 выступает прямая поддержка отношений, которые позволяют отслеживать связи между объектами. Это заданные системой отно-

шения, например, когда один объект содержит другой, «носимый», объект, но разработчик может добавить и свои собственные отношения (например, любви или ненависти между людьми) либо отслеживать, какие персонажи встречались друг с другом.

На переднем крае исследований находятся разработки MIT Media Lab, в частности система Metafor [3]. Metafor может перевести фразу «Pacman is a yellow character who eats dots» в заготовку исполняемого кода, в котором есть объект Pacman со свойством color=yellow класса character и метод класса eat (dot). С помощью Metafor можно создать игру, способную «понять» следующий текст:

There is a brown room with lights in every corner, and a treasure chest in the center. If someone opens the treasure chest, the lights will all turn off.

На сайте [4] предлагается подход, заключающийся в использовании только «правильного» подмножества ЕЯ, которое называется sEnglish.

Следующая теорема – это код на sEnglish, который компилируется в исполняемый код на MATLAB:

Pythagoras' Theorem

Let T be a 'triangle'. Let T have 'side lengths' denoted by a, b, c. Let T have 'angles' denoted by alpha, beta, gamma. If gamma of T is equal to  $\sim 90^\circ$ , then  $c^2 = a^2 + b^2$  within measurement tolerances.

Определение ЕЯ в данной статье в зависимости от целей разработчиков языков, приближенных к ЕЯ, будет либо слишком узким, либо слишком широким. Приведенную теорему Пифагора поймет почти любой, владеющий английским, и признает, что это язык общения обычных людей между собой, а не специалистов-математиков. Если исключим из нее сокращения и спецсимволы, то придем к пресловутому КОБОЛУ, а если будем использовать их по максимуму, то превратим текст в слабочитаемый и текстом на ЕЯ он уже считаться не сможет.

Иногда программа на sEnglish становится излишне многословной:

Attributes of a phase plot.

A 'phase plot' has the following properties: its 'highest frequency' that is a number array, its 'lowest frequency' that is a number array and its 'plot object handle' that is a number array.

А иногда является образцом того, как должны выглядеть программы на ЕЯ:

If U\_ is 'smc01-control', then do the following. Define surface weights Alpha as "[0.5, 0.5]". Initialise matrix Phi as a 'unit matrix'. Define J as the 'inertia matrix' of Spc01. Compute matrix J2 as the inverse of J. Compute position velocity error Ve and angular velocity error Oe from dynamical state X, guidance reference Xnow. Define the joint sliding surface G2 from the position velocity error Ve and angular velocity error Oe using the surface weights Alpha. Compute the smoothed sign function SG2 from the joint sliding surface G2 with sign threshold 0.01. Compute special dynamical force F from dynamical state X and surface weights Alpha. Execute "Diffdot=zeros(6,1);". Compute control torque T and control force U from matrix J2, surface weights Alpha, special dynamical force F, smoothed sign function SG2 and Diffdot. Finish conditional actions.

Данный отрывок, безусловно, можно считать текстом на ЕЯ, только естественен он для ограниченного круга людей – физиков и инженеров. Если бы их попросили максимально понятно описать последовательность действий для своего коллеги, они использовали бы именно этот язык. Таким образом, ЕЯ включает в себя множество различных языков – не только языков разных народов, но и языков различных специальностей.

sEnglish обладает той важной особенностью, что в него можно включать отрывки кода на системе более низкого уровня – MATLAB. Это позволяет иногда сократить запись.

В статье [5] рассматривается возможность перевода описаний на ЕЯ в регулярные выражения.

Регулярные выражения (регэкспы) зарекомендовали себя как чрезвычайно мощный и универсальный формализм, который проник во все типы программ: от электронных таблиц до баз данных. Однако даже многие программисты не до конца понимают, как работают регэкспы. Таким образом, способность автоматически генерировать регулярные выражения из естественного языка была бы полезна во многих ситуациях.

Когда Н. Кушман представил доклад в соавторстве с Р. Барзилэй, он попросил группу информатиков записать регулярное выражение, соответствующее достаточно простому поиску в тексте. Когда он показал ответ и попросил поднять руки тех, кто ответил правильно, подня-

лось только несколько рук. Таким образом, система может быть полезна для зрелых программистов, но она также может позволить обычным пользователям, скажем, электронных таблиц и текстовых процессоров задавать сложный поиск с использованием ЕЯ.

Парсеры осуществляют синтаксический анализ текста. На вход парсера подается текст, на выходе строится дерево разбора. В статье [6] представлен метод для автоматической генерации парсеров из спецификаций форматов входных файлов на английском языке. Парсеры входных данных предназначены для выяснения, какие части файла содержат определенные типы данных. Без такого парсера файл – это просто случайный набор нулей и единиц. Для отображения релевантных явлений естественного языка и перевода спецификации на английском языке в дерево спецификации, которое затем транслируется в парсер на C++, используется порождающая модель Байеса.

Исследователи Массачусетского технологического института проверили генератор парсеров более чем на ста примерах, отобранных из текстов задач олимпиад по программированию АСМ, которые включали спецификации файлов для каждой задачи. Генератор был в состоянии производить рабочие парсеры примерно для 80 % спецификаций, и в остальных случаях изменение слова или двух в спецификации обычно порождало работающий парсер.

Значительным достижением представляется работа с ЕЯ в системе Wolfram Alpha, описанная на сайте [7]. К примеру, данная система способна интерпретировать следующие предложения: draw a translucent tiffany blue sphere and a red cone, 20 random integers between -5 and 5, positions of negative numbers in t. К сожалению, она является платной и с закрытым исходным кодом.

В статье [8] рассматривается попытка программирования пространственных алгоритмов на ЕЯ. Входными данными для предлагаемой системы служит естественно-языковое описание алгоритма обработки пространственных данных, а выходными – объектно-ориентированный программный код, который должен быть скомпилирован и выполнен. Предложены и оценены два подхода. Первый основан на текстовом сопоставлении с образцом: для каждого предложения выбирается наилучшим образом подходящий шаблон, а объекты и методы создаются в соответствии с этим шаблоном. Второй подход преобразует текст в логическую форму, подвергающуюся ряду преобразований для получения результирующего кода. Для выполнения этих преобразований на каждом шаге используется ряд эвристических правил. Затем требуется дополнительный проход для обработки ссылок, чтобы найти одинаковые объекты, методы и переменные среди инструкций кода. Результаты предварительного исследования показывают, что с помощью системы программирования на ЕЯ в интерактивном режиме, где пользователь вручную редактирует сгенерированный код, можно значительно повысить производительность кодирования. Тем не менее в настоящее время точность полностью автоматизированного, т. е. неинтерактивного, режима генерации кода по-прежнему слишком низка, чтобы быть полезной.

## 2. Новые подходы к разработке естественных языков программирования

В статье [9] рассматривается применение теории онтологий, концептуальных графов, а также теории языков программирования к разработке теоретических основ программирования на ЕЯ, которое было в последние годы использовано для создания документов на ЕЯ для интеллектуальных агентов и читателей-людей. Проведенный анализ показывает три преимущества программирования на ЕЯ. Во-первых, это концептуализированное программирование, которое позволяет разработчикам писать программы с меньшим количеством ошибок благодаря ясности представления кода и вынужденному структурированию данных. Во-вторых, программирование на ЕЯ может помочь программированию всех важных абстракций для роботов: события, действия и модели мира могут быть созданы с помощью предложений. В-третьих, программирование на ЕЯ может использоваться для публикации исследователями документов на ЕЯ, т. е. документов по теории и процедурам управления роботами. Данная теоретическая работа также определяет большой класс интеллектуальных агентов, которые могут читать такие документы. Все это позволяет пользователям-людям и агентам иметь общее понимание того, как работают прикладные системы.

В статье [10] рассматривается модель для формализованного представления знаний «концептуальные графы». Человеческие знания можно смоделировать с помощью наборов концепту-

альных графов, которые могут иметь модальности прошлого, настоящего и будущего, или косвенной речи. Концептуальные графы могут быть и абстракцией более сложной концептуальной модели, описанной графами. Некоторые концептуальные графы основаны на восприятии окружающего физического мира и являются прямыми абстракциями процессов восприятия.

ЕЯ обладает богатой семантикой, но, к сожалению, его неформальные выразительные возможности часто ошибочно принимаются за простую неточность. Поскольку полные парсеры английского языка еще недоступны, люди не представляют возможным использовать английский язык непосредственно как средство программирования компьютеров. Тем не менее в статье [11] показывается, что описания процедур на английском языке часто содержат семантику программирования – языковые особенности, которые могут быть легко отображены в конструкции языка программирования. Некоторые лингвистические особенности могут даже вдохновить на новые способы мышления о задании программы. Далекие от того, чтобы быть безнадежно неоднозначными, ЕЯ основываются на важных принципах коммуникации, которые могут быть использованы, чтобы сделать взаимодействие человека с компьютером более естественным.

Для демонстрации возможности программирования на ЕЯ в статье [12] анализируются случаи из числа самых трудных: шаги и циклы. Рассматривается корпус описаний на английском языке, используемых в качестве заданий по программированию, и разрабатываются некоторые методы для отображения лингвистических конструкций на программные структуры, которые являются программной семантикой.

В статье [13] высказывается мнение, что улучшенные технологии обработки языка, диалог со смешанной инициативой и программирование на примерах могут сделать возможным программирование на ЕЯ сейчас, хотя это не было возможным в прошлом.

*Улучшенные технологии обработки языка.* В то время как полное понимание ЕЯ по-прежнему остается вне досягаемости, есть шанс, что недавние улучшения в надежном грамматическом разборе с широкой областью применения, семантически-управляемом синтаксическом разборе, поверхностном парсинге (chunking) и успешном развертывании управляющих систем на ЕЯ могут обеспечить частичное понимание, достаточное для начала разработки практической системы.

*Диалог со смешанной инициативой.* Принцип «пользователь будет просто читать код вслух» заменяется на «пользователь и система должны разговаривать о программе». Система должна пробовать интерпретировать то, что пользователь говорит о программе, а затем спросить пользователя о том, что она не понимает, предоставить недостающую информацию и устранить недоразумения.

*Программирование на примерах.* Принимается методология «покажи и расскажи», которая объединяет описания на естественном языке с конкретными демонстрациями на примере. Иногда легче продемонстрировать то, что вы хотите, чем описать словесно. Пользователь может сказать системе: «Вот то, что я хочу». Система, в свою очередь, может проверить свое понимание: «Это то, что вы имеете в виду?» Это сделает систему более отказоустойчивой в тех случаях, когда язык не может быть непосредственно понят, а в случае полной неудачи более сложных методов пользователь может ввести код на низкоуровневом языке программирования.

MOOIDE [14] является интерфейсом, позволяющим начинающим пользователям запрограммировать среду MOO, используя ЕЯ. Программирование MOO включает разнообразные задачи, такие как создание объектов и их состояний, назначение глаголов-действий объектам и программирование поведения, которое изменяет состояния объектов и генерирует сообщения. После того как MOO запрограммирован, другие пользователи могут взаимодействовать с объектами в развлекательных или образовательных целях.

Для того чтобы сделать программирование MOO проще и доступнее для начинающих программистов, интерфейс на ЕЯ позволяет пользователям описывать различные задачи программирования MOO на английском языке, а именно добавление объектов, свойств объектов, состояний и отношений между объектами, а также глаголы, с помощью которых осуществляется доступ к поведению объектов в MOO. Английский язык может использоваться для описания инструкций принятия решений, циклов, условий и других типичных конструкций программирования.

Более ранние системы были сосредоточены на решении проблем синтаксического анализа в программировании, однако этим системам не хватало общеизвестных знаний. MOOIDE

привносит в программирование на ЕЯ оперирование на основе здравого смысла в дополнение к синтаксическому разбору. Рассуждения на основе здравого смысла позволяют MOOIDE автоматически поддерживать типичные свойства объектов, а также допустимые эффекты глаголов. Подобное дополнение в программирование на ЕЯ с помощью рассуждений поможет сделать его значительно более интуитивным для начинающих программистов.

В статье [15], где описывается Ubiquity – эксперимент Mozilla Labs по разработке естественно-языкового интерфейса для браузера, высказывается важная мысль: *«Если синтаксис языка является слишком ограничительным или, что еще хуже, конфликтует с природной интуицией пользователя о его языке, язык моментально перестает быть естественным независимо от того, насколько похожи ключевые слова или грамматика на таковые в естественном языке»*.

### 3. Язык CLIPS

Рассмотрим язык CLIPS, который является декларативным, т. е. должен облегчать программирование некоторых задач по сравнению с императивными языками. Основными элементами CLIPS являются факты, структуры, правила и функции.

Пример определения структуры:

```
(deftemplate personal-data
  (slot name)
  (slot age)
  (slot weight)
  (multislot date_of_birth)
)
```

где slot – это поле, принимающее одно значение, а multislot – поле, принимающее список значений.

Определение фактов:

```
(deffacts trouble_shooting
  (car_problem (name ignition_key) (status on))
  (car_problem (name engine) (status wont_start))
  (car_problem (name headlights) (status work))
)
```

В этих фактах используется структура car\_problem с полями name и status. Ясно, что синтаксис совершенно не похож на синтаксис ЕЯ. Факты можно добавлять, отображать и удалять.

Правило для псевдокода

```
IF the animal is a duck
THEN the sound made is quack
```

выглядит следующим образом:

```
(defrule duck
  (animal-is duck)
=>
  (assert (sound-is quack)))
```

И опять код достаточно далек от псевдокода.

Еще пример правила:

```
(defrule not-yellow-red
  (light ?color&~red&~yellow)
=>
  (printout t "Go, since light is "
    ?color crlf))
```

Пример функции:

```
(deffunction hypotenuse (?a ?b)
  (sqrt (+ (* ?a ?a) (* ?b ?b))))
```

CLIPS использует синтаксис Лиспа, поэтому подобные функции сложно воспринимать даже программистам, использующим другие языки. Учитывая, что семантика CLIPS так же бедна, как и синтаксис, можно сделать вывод, что он не менее далек от программирования на ЕЯ, чем любой другой современный язык программирования высокого уровня.

#### 4. Русскоязычная литература

На русском языке также имеется литература, близкая по тематике к программированию на ЕЯ. В частности, следует отметить монографию О.Е. Елисеевой «Естественно-языковой интерфейс интеллектуальных систем» [16]. Она состоит из трех частей:

1. Обобщенное рассмотрение языка и естественно-языкового интерфейса.
2. Конкретные синтаксические и семантические особенности русского языка, а также свойства, общие для всех языков.
3. Рассмотрение реализации конкретной системы.

Первая часть не представляет особого интереса для специалиста, неплохо владеющего базовыми знаниями в области обработки ЕЯ и желающего лишь ознакомиться с новыми исследованиями по программированию на ЕЯ. В ней изложены общие вопросы лингвистики и психологии, а также темы, которые находятся на слишком высоком уровне абстракции по сравнению с уровнем компьютерных языков программирования. К примеру, рассматривается «поэтическая, или эстетическая, функция языка».

Во второй части теория становится ближе к практике. Данная часть включает в себя следующие темы:

- уровни изучения текста от отдельных морфем до полной картины сведений о мире;
- алгоритмы анализа и синтеза естественно-языковых текстов от морфологического анализа до построения базы знаний;
- морфология, т. е. классификация слов по частям речи и деление слов на морфемы;
- синтаксическая структура предложения;
- семантический язык и его связь с синтаксическим;
- семантический анализ.

Все темы изложены настолько подробно, что вполне могут послужить основой для разработки алгоритма на псевдокоде. Достаточное внимание ко всем рассмотренным вопросам позволит создать полноценный язык программирования.

В третьей части кратко представлены компоненты конкретных систем общения с пользователем на ЕЯ, а также принципы построения интерфейса интеллектуальных систем.

#### Заключение

Программирование на ЕЯ находится на том этапе, когда написаны первые прототипы систем программирования и открываются перспективные пути исследований. Полученные положительные результаты свидетельствуют о том, что вскоре должен произойти переход к следующему этапу – разработке систем промышленного программирования на языке, приближенном к естественному. На первом этапе главным был вопрос «Насколько реализуемы данные системы?», на втором таким вопросом будет «Насколько они практичны?». Разработка языка – не только научная, но и инженерная, и психологическая задача. Написание программ на языке должно быть удобным для программиста и давать положительный экономический эффект.

#### Список литературы

1. Dijkstra, E.W. On the foolishness of «natural language programming» / E.W. Dijkstra [Electronic resource]. – 1978. – Mode of access : <http://cs.utexas.edu/users/EWD/transcriptians/EWD06XX/EWD667>. – Date of access : 02.02.2015.

2. Nelson, G. Natural language, semantic analysis and interactive fiction / G. Nelson // Inform7 web site [Electronic resource]. – 2005. – Mode of access : <http://inform7.com/learn/documents/WhitePaper.pdf>. – Date of access : 15.02.2015.
3. Liu, H. Metafor : Visualizing Stories as Code / H. Liu, H. Lieberman // Proc. of the 10<sup>th</sup> Intern. Conf. on Intelligent User Interfaces. – NY, 2005. – P. 305–307.
4. Veres, S.M. Sysbrain: natural language programming / S.M. Veres, A. Tsourdos // Sysbrain [Electronic resource]. – 2008. – Mode of access : <http://sysbrain.org>. – Date of access : 08.02.2015.
5. Kushman, N. Using Semantic Unification to Generate Regular Expressions from Natural Language / N. Kushman, R. Barzilay // Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, Proc. – Cambridge, 2013. – P. 826–836.
6. From Natural Language Specifications to Program Input Parsers / T. Lei [et al.] // The 51<sup>st</sup> Annual Meeting of the Association for Computational Linguistics (ACL 2013). – Sofia, 2013. – P. 1294–1303.
7. Programming with natural language is actually going to work [Electronic resource]. – 2010. – Mode of access : <http://blog.wolfram.com/2010/11/16/programming-with-natural-language-is-actually-going-to-work>. – Date of access : 03.02.2015.
8. Galitsky, B. Programming Spatial Algorithms in Natural Language / B. Galitsky, D. Usikov // AAAI Workshop Technical Report WS-08-11. – Palo Alto, 2008. – P. 16–24.
9. Veres, S.M. Theoretical foundations of natural language programming and publishing for intelligent agents and robots / S.M. Veres // TAROS 2010 Proc. – Southampton, 2010. – P. 292–299.
10. Veres, S.M. Documents for intelligent agents in English / S.M. Veres, L. Molnar // Artificial Intelligence and Applications 2010 Conf. Proc. – Innsbruck, 2010. – P. 10.
11. Liu, H. Toward a Programmatic Semantics of Natural Language / H. Liu, H. Lieberman // IEEE Symposium on Visual Languages and Human-Centric Computing 2004 Proc. – Cambridge, 2004. – P. 281–282.
12. Mihalcea, R. NLP (Natural Language Processing) for NLP (Natural Language Programming) / R. Mihalcea, H. Liu, H. Lieberman // Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science. – 2006. – Vol. 3878. – P. 319–330.
13. Lieberman, H. Feasibility studies for programming in natural language / H. Lieberman, H. Liu // Kluwer Academic Publishers. – Dordrecht, 2005. – 16 p.
14. Ahmad, M. MOOIDE : Natural Language Interface for Programming MOO Environment / M. Ahmad. – Massachusetts Institute of Technology, 2008. – 70 p.
15. How natural should a natural interface be? Michael Yoshitaka Erlewine blog [Electronic resource]. – 2009. – Mode of access : <http://mitcho.com/blog/projects/how-natural-should-a-natural-interface-be>. – Date of access : 25.01.2015.
16. Елисеева, О.Е. Естественно-языковой интерфейс интеллектуальных систем : учеб. пособие / О.Е. Елисеева; под ред. В.В. Голенкова. – Минск : БГУИР, 2009. – 151 с.

Поступила 20.02.2015

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: dsblizzard@gmail.com*

**E.G. Luttsev**

## **PROGRAMMING IN NATURAL LANGUAGE: PUBLICATIONS REVIEW**

Paper addresses a number of scientific papers devoted to the issues of programming languages close to natural languages. Description of classical approaches to the design of natural programming languages and new approaches that have made the development of these languages practical is given. One of the high-level languages – CLIPS – is compared with natural programming languages. A monograph in Russian devoted to the natural language interfaces is reviewed.