

# МОДЕЛИ КОГНИТИВНЫХ ПРОЦЕССОВ (Вычислительные системы)

1998 год

Выпуск 164

УДК 519.517.12

## АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ ДЛЯ РЕАЛИЗАЦИИ ЯЗЫКОВ СПЕЦИФИКАЦИЙ ЗАДАЧ<sup>1</sup>

А.А.Москвитин

### В в е д е н и е

Одной из самых важных проблем информатики является точная постановка задачи из некоторой предметной области и ее адекватное решение на компьютере. Иначе говоря, возникает потребность в разработке специальных языков спецификаций задач, ориентированных на пользователя не являющегося профессионалом в программировании, но точно знающего чего он хочет от компьютера. Решить данную проблему можно различными путями (например, подвергнуть пользователя многочасовому изучению теоретического материала по тематике задачи и выработке навыков работы на компьютере). Иной подход к решению данной проблемы описан в работах [1,2]. Для реализации данного подхода и предлагается описываемый ниже проект программной системы.

В работе [3] описаны основные принципы построения программного обеспечения и технология работы при создании программного обеспечения, реализующего указанный в заголовке проект. Однако для реализации такой технологии требуется некоторый компьютерный инструментарий. На наш взгляд,

---

<sup>1</sup>Работа выполнена при поддержке гранта РФФИ № 96-06-80570

наиболее полно отвечает требуемым свойствам инструментальная система Borland C++ Builder. Именно на ее основе и предполагается реализация программной системы для реализации языков спецификаций задач, ориентированных на пользователя.

Выбор данной инструментальной системы осуществлен по следующим соображениям:

- необходимость использования всех современных наработок в области программного обеспечения, включая визуальные средства, объектно-ориентированное программирование и другие возможности, которые присутствуют в языках C++ и Delphi;

- возможность применения новейшего стиля программирования, позволяющего ускорить процесс разработки качественного программного обеспечения;

- удобство и простота использования инструментария, в том числе, средств визуального и объектно-ориентированного программирования;

- возможность неограниченного распараллеливания работ между группами разработчиков проекта и простота объединения созданных частей в единое целое;

- максимальный учет возможностей конкретного вычислительного устройства и его операционной среды, в сочетании с относительной простотой адаптации созданного программного обеспечения применительно к другим вычислительным устройствам и операционным средам;

- возможность поддержки работы системы как в автономном, так и в сетевом режимах, с одновременным применением современных средств хранения, преобразования и обмена информацией.

Это далеко не полный перечень тех возможностей Borland C++ Builder, который позволяет сделать выбор в его пользу как инструмента создания системы для реализации языков спецификаций задач.

Несколько слов о структуре создаваемого программного обеспечения для реализации языков спецификаций задач. Если внимательно посмотреть на работы [1,2], то можно заметить, что процессу решения пользователем своей задачи на компьютере должен предшествовать этап выяснения его интеллектуальных возмож-

ностей и их сочетания с его интеллектуальными потребностями. В этом случае может оказаться две возможности:

- 1) интеллектуальных ресурсов пользователя достаточно для адекватной постановки и решения задачи на компьютере;
- 2) интеллектуальные возможности пользователя не соответствуют (недостаточны) интеллектуальным потребностям решаемой задачи.

В первом случае пользователю имеет смысл попробовать сформулировать свою задачу на языке спецификаций и решить ее на компьютере. Во втором — лучше даже не браться за это самому, а, в лучшем случае, обратиться за помощью к специалисту по постановкам задач или же привести свои интеллектуальные возможности в соответствие с интеллектуальными потребностями решаемой задачи.

Согласно сказанному, все программное обеспечение языков спецификаций задач распадается на две взаимно связанные (и в тоже время независимые) части. Первая часть отвечает за тестирование пользователя на предмет выяснения его способностей к решению задачи (его интеллектуальных возможностей). Эта же часть программного обеспечения одновременно служит, в некоторой степени, вспомогательным средством обучения постановке задачи. Вторая часть — предоставляет возможность пользователю сформулировать и решить свою задачу.

## 1. Определение интеллектуальных ресурсов пользователя

Напомним, какие задачи решаются на этапе тестирования пользователя на предмет выявления его интеллектуальных ресурсов (соответствия его интеллектуальных потребностей его интеллектуальным возможностям) [1].

Интеллектуальный ресурс (данного человека)  $p$  относительно (данной аксиоматической системы)  $S$  есть тройка  $\text{res}(p, S) = (m_1(p, S), m_2(p, S), m_3(p, S))$  натуральных чисел  $m_1(p, S)$ ,  $m_2(p, S)$ ,  $m_3(p, S)$  таких, что:

—  $m_1(p, S)$  — наибольшая длина доказательств в  $S$ , все еще имеющих достаточно высокую (заранее фиксированную) балльную оценку степени убедительности для данного человека  $p$ ;

—  $m_2(p, S)$  — наибольшая длина последовательностей из символов алфавита языка системы  $S$ , все еще имеющих достаточно высокую (заранее фиксированную) балльную оценку (субъективной уверенности  $p$  в) безошибочной распознаваемости их  $p$  как формул (или не формул) языка системы;

—  $m_3(p, S)$  — то же, что и  $m_2(p, S)$ , но применительно к термам.

Таким образом, проблема измерения интеллектуального ресурса пользователя  $p$  (относительно  $S$ ) сводится к следующим трем задачам [2]:

1) исследовать вопрос о влиянии длины логических выводов в  $S$  на степень убедительности их для  $p$ ;

2) исследовать вопрос о влиянии длины последовательности символов на уверенность признания данным человеком  $p$  этой последовательности формулой (или не формулой) языка системы  $S$ ;

3) исследовать вопрос о влиянии длины последовательности символов на уверенность признания человеком  $p$  этой последовательности термом (или не термом) языка системы  $S$ .

В данной работе рассматривается проект программной системы, позволяющей решить третью из перечисленных выше задач, а именно исследовать вопрос о влиянии длины последовательности символов на уверенность признания человеком  $p$  этой последовательности термом (или не термом) языка системы  $S$ .

Заметим, что после того, как все указанные выше задачи успешно решены у пользователя появляется реальная возможность приступить к решению своей задачи.

Таким образом, программное обеспечение, поддерживающее языки спецификаций задач распадается на две самостоятельные, но взаимно зависимые части. В первой части выявляются интеллектуальные ресурсы пользователя  $p$ , а во второй — самим пользователем осуществляется постановка и решение своей задачи с использованием языка спецификаций.

Решение третьей задачи первой части программной системы PLAST начинается с регистрации пользователя. Это необходимо сделать для того, чтобы в дальнейшем можно было идентифицировать его при выдаче рекомендаций о возможности приступить

к решению задачи. При регистрации в системе PLAST пользователь указывает свои фамилию, имя и отчество, а также сообщает основные сведения о предметной области, в которой предполагается решение задачи. Здесь же он определяет все элементы, необходимые для построения термов (а в дальнейшем формул) или не термов, задает сигнатуру.

В настоящей версии этот процесс осуществляется через специальные "меню" выбора, а в дальнейшем предполагается перевести его в произвольную форму.

Для реализации первой части программной системы PLAST разработано три модуля, которые отвечают соответственно за следующие виды работ. Первый модуль осуществляет регистрацию пользователя  $p$  в системе с выдачей формы 1 (рис.1) и организует ведение базы данных. Второй модуль осуществляет генерацию массивов конструкций (термов и не термов) по заданной пользователем аксиоматической системе  $S$ . Третий модуль отвечает за предъявление пользователю  $p$  (форма 2) групп конструкций (термов или не термов) для распознавания и фиксирует ответы пользователя.

Форма 1	
Фамилия: Иванов	
Имя: Иван	Отчество: Иванович
Предметная область: математика	
Сигнатура:  $\sigma = \langle f_1^2, f_2^2, f_3^0, f_4^0 \rangle$  $m_1$ — параметр длины	

Рис. 1

По результатам регистрации ведется специальный журнал, связанный с базой данных об испытаниях, проводимых с конкретным пользователем. В дальнейшем эти данные используются при постановке пользователем  $p$  своей задачи.

Далее пользователю  $p$  предлагается пройти два этапа по выявлению его способностей к распознаванию термов (и формул, имеется в виду логических) по заданной самим пользователем сигнатуре и при автоматическом увеличении длин термов (и формул). Указанный процесс осуществляется следующим образом.

После регистрации пользователя  $p$ , система автоматически генерирует правильные и неправильные термы (и формулы) длины  $n$  в терминах его предметной области. Затем термы (или формулы) и конструкции, похожие на термы (или формулы), датчиком случайных чисел выбираются из соответствующих массивов и предъявляются пользователю для опознания через форму 2 (рис.2).

## Форма 2


<p><b>Сигнатура</b></p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math>&lt; + . ' 0 &gt;</math> </div> <p><b>Является ли конструкция термом?</b></p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>1. <math>x + y</math></p> <p>2. <math>xy</math></p> <p>3. <math>x^2 + 8</math></p>            <p>10. <math>(x + y)</math></p> </div>	<p><b>Таймер</b></p> <div style="text-align: center; margin: 10px 0;">  </div> <p><b>Ответ</b></p> <div style="margin-top: 20px;"> <p>1. <input checked="" type="checkbox"/> <b>V</b>    V — да</p> <p>2. <input type="checkbox"/></p> <p>3. <input type="checkbox"/></p> <p style="margin-left: 100px;"><input type="checkbox"/></p> <p style="margin-left: 100px;"><input type="checkbox"/></p>     <p>10. <input type="checkbox"/></p> </div>
---	---

Рис. 2

Выдача пользователю  $p$  термов (или формул) или конструкций, напоминающих термы (или формулы), заданной сигнатуры и определенной длины  $n$  осуществляется группами по десять штук на экран и одновременно включается таймер. В левом верхнем углу пользователю высвечивается вспомогательная информация, необходимая ему для принятия правильного решения при опознании. Условно эта информация названа сигнатурой. Справа от каждой конструкции, имеется окошко в котором пользователь отмечает радиокнопкой правильный терм (или формулу).

По окончании времени, отведенному для распознавания при включенном таймере, пользователю  $p$  предъявляется очередная порция конструкций на распознавание. При 100% распознавании термов первоначальной длины  $n$  пользователю выдаются термы длины  $n + 1$  и так продолжается до тех пор пока пользователь в состоянии правильно распознавать термы (или формулы) или же пока не закончится процесс тестирования. Далее, согласно методике установления ресурсов пользователя изложенной в [2], пользователю  $p$ , рассуждающему в рамках аксиоматической системы  $S$ , приписывается интеллектуальный ресурс  $res(p, S) = (m_1(p, S), m_2(p, S), m_3(p, S))$ .

## 2. Схема постановки задачи пользователем $p$

Процесс постановки пользователем  $p$  задачи в терминах предметной области и ее решение на компьютере возможно только тогда, когда пользователь обладает достаточным интеллектуальным ресурсом, что, в свою очередь, выясняется на предыдущем этапе тестирования. Иначе говоря, его интеллектуальных возможностей должно быть достаточно для удовлетворения его интеллектуальных потребностей при постановке и решении определенной задачи из некоторой предметной области, в которой он проходил тестирование. Если это так, то наступает этап формирования задачи.

В этом случае он в диалоге с программной системой формулирует свою задачу в точных терминах языка спецификаций задач, пользуясь при этом терминами из некоторой предметной области, к которой относится его задача. Процесс формулировки задачи

может быть многоэтапным и заканчивается автоматическим решением задачи на компьютере.

Методика формулировки задачи в терминах предметной области описана в работе [4] и продемонстрирована на примере решения задачи формирования технического задания на выполнение определенной работы по созданию программного обеспечения в указанных ограничениях.

Если же интеллектуальных ресурсов пользователя  $p$  недостаточно для решения задачи, то в этом случае ему предоставляется возможность описать свою задачу в терминах предметной области для последующего перевода ее на язык спецификаций задач специалистом в этой области.

### З а к л ю ч е н и е

Таким образом, появляется возможность пользователю, не являющемуся профессионалом в программировании, но овладевшим методикой языков спецификаций задач и являющемуся профессионалом в своей области, качественно решать свои задачи на компьютере, не прибегая к услугам программистов.

В настоящий момент разработано программное обеспечение, поддерживающее этап тестирования пользователя на предмет правильного распознавания им термов длины  $n$  в заданной аксиоматической системе  $S$ . Термы задаются в линейной форме. Требуется доработать программное обеспечение до общепринятого вида изображения термов и выдачи их на экран монитора, для предъявления их пользователю.

### Л и т е р а т у р а

1. КАЗАКОВ Е.В., МОСКВИТИН А.А., САМОХВАЛОВ К.Ф. Проект разработки языков спецификаций задач, ориентированных на пользователя // Модели когнитивных процессов. — Новосибирск, 1997. — Вып. 158: Вычислительные системы. — С. 63–94.

2. КАЗАКОВ Е.В., МОСКВИТИН А.А., САМОХВАЛОВ К.Ф. Установление ресурсов пользователей // Измерение и модели когнитивных процессов. — Новосибирск, 1998. — Вып. 162: Вычислительные системы. — С. 41–57.



3. МОСКВИТИН А.А. Об одном способе организации программного обеспечения в технологическом комплексе решения логических задач //Модели когнитивных процессов. — Новосибирск, 1997. — Вып. 158: Вычислительные системы. — С. 95-1-6.

4. МОСКВИТИН А.А. Среда спецификационной деятельности СИГМА-ТЗ //Теория вычислений и языки спецификаций. — Новосибирск, 1995. — Вып. 152: Вычислительные системы. — С. 76-94.

Поступила в редакцию  
30 ноября 1998 года