

Дрождин В.В., Шалаев А.А. Совершенствование механизма самомодификации адаптивной информационной системы. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XVI Междунар. научно-техн. конф. – Пенза: ПДЗ, 2016. – С. 144-152.

УДК 004.436.4

СОВЕРШЕНСТВОВАНИЕ МЕХАНИЗМА САМОМОДИФИКАЦИИ АДАПТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

В.В. Дрождин, А.А. Шалаев

IMPROVEMENT OF THE SELF-MODIFICATION PROCESS IN AN ADAPTIVE INFORMATION SYSTEM

V.V. Drozhdin, A.A. Shalaev

Аннотация. Предложен способ передачи фактических параметров в подзадачи и методы получения результатов в виде значений выражений, расширяющий возможности системы по согласованию параметров программных компонентов. Это позволяет уменьшить объем семантических описаний задач в системе и повысить быстродействие подсистемы самомодификации адаптивной информационной системы.

Ключевые слова: адаптация, самомодификация, информационная система, задача, метод решения, семантическое описание, язык разметки.

Abstract. The article proposes a way of passing parameters to methods and subtasks as parameter values or as expressions. This expands the capabilities of the system to coordinate parameters between program modules. This approach also allows reducing the amount of semantic descriptions and improving the performance of the self-modification subsystem in an adaptive information system.

Keywords: adaptation, self-modification, information system, task, method, semantic description, markup language

Разработанные в [1–3] организация и алгоритмы функционирования подсистемы самомодификации адаптивной информационной системы (АДИС) на основе семантического и конструктивного описаний алгоритмов позволяют системе приобрести принципиальную возможность формирования корректных, надежных и эффективных алгоритмов решения задач в различном контексте из набора методов, известных системе. Это позволяет АДИС автоматически (без участия человека) поддерживать высокую адекватность изменяющейся внешней среде и внутренней организации системы. Однако предложенная структура семантического описания алгоритмов на основе однозначного соответствия входов и выходов задачи, подзадач и методов их решения приводит к громоздкости и большому числу уровней описаний процесса решения задачи, разрешимой системой, что существенно снижает эффективность функционирования подсистемы самомодификации. Поэтому необходимо усовершенствовать способ согласования параметров таким образом, чтобы он не требовал больших изменений подсистемы самомодификации АДИС, не предъявлял дополнительных требований к пользователям, но повышал эффективность механизма самомодификации системы.

Для этого рассмотрим формальное описание программной реализации решения задачи нахождения оптимальной цены нового товара:

$$\text{task}(1, In = (k_{max}, price_{max}, e, cost), Out = (price), q = ((k_{max} - \frac{k_{max}}{price_{max}} * price) * price - e + cost * (k_{max} - \frac{k_{max}}{price_{max}} * price) = max), c = (price \in$$

$[price_1, price_2]$, где $price_1, price_2$ – корни уравнения $(k_{max} - \frac{k_{max}}{price_{max}} * price) * price - e + cost * (k_{max} - \frac{k_{max}}{price_{max}} * price) = 0$, (: method₁);

method(1, $In^\mu = (k_{max}, price_{max}, e, cost)$, $Out^\mu = (price)$, $f = ((k_{max} - \frac{k_{max}}{price_{max}} * price) * price - e + cost * (k_{max} - \frac{k_{max}}{price_{max}} * price) = max)$, $c^\mu =$

$(price \in [price_1, price_2]$, где x_1, x_2 – корни уравнения $(k_{max} - \frac{k_{max}}{price_{max}} * price) * price - e + cost * (k_{max} - \frac{k_{max}}{price_{max}} * price) = 0$), (: p₁);

p₁ = (

// Исходные данные

coef_a = k_max / price_max;

coef_b = k_max + cost * k_max / price_max;

coef_c = -(e + cost * k_max);

// Расчет

discriminant = coef_b * coef_b - 4 * coef_a * coef_c;

if (discriminant < 0)

price1 = price2 = -1;

else if (discriminant > 0)

{

sqrt = Sqrt(discriminant);

price1 = (-coef_b + sqrt) / (2 * coef_a);

price2 = (-coef_b - sqrt) / (2 * coef_a);

}

else

price1 = price2 = (-coef_b + Sqrt(discriminant)) / (2 * coef_a);

// Результат

price = (x1 + x2) / 2;

)

где k_{max} – максимально возможный объем продаж аналогичного товара;

$price_{max}$ – максимальная цена аналогичного товара;

e – начальные затраты;

$cost$ – себестоимость единицы товара;

$a = \frac{k_{max}}{price_{max}}$ – коэффициент эластичности объема продаж от цены товара;

$price$ – цена единицы товара;

$k = k_{max} - a * price$ – ожидаемый объем продаж товара при определенной цене;

$task, method$ и p_1 – соответственно семантическое описание, метод и алгоритм (программный код) решения задачи.

Из приведенного описания видно, что решение задачи заключается в решении квадратного уравнения, исходные данные и результат которого формируются из параметров, имеющих интерпретацию в предметной области (в терминах решаемой задачи). Вследствие наличия предобработки и постобработки в рамках ранее разработанного подхода данная задача должна быть декомпозирована на задачу и три подзадачи:

- 1) выделение предобработки в отдельную подзадачу, осуществляющую проблемную ориентацию абстрактной задачи;
- 2) подзадача, осуществляющая решения абстрактной задачи (решение квадратного уравнения);
- 3) выделение постобработки в отдельную подзадачу, осуществляющую проблемную ориентацию результата решения абстрактной задачи;
- 4) задача, организующая вычисление оптимальной цены нового товара путем последовательного вызова подзадач 1-3.

Подзадачи 1-3 являются простыми задачами, для решения которых имеется определенный способ решения. Простая задача осуществляет интерпретацию абстрактных методов на семантическом уровне. Заметим, что только подзадача 2 соответствует одной из классических задач математики – решению квадратного уравнения, а остальные вызваны ограниченностью соответствия параметров.

Задача 4 будет представлена только семантическим описанием на языке разметки, определяющем последовательность вызова подзадач и согласование входных параметров и результатов вычислений.

Так как параметры, передаваемые из подзадачи более высокого уровня в подзадачу более низкого уровня и из простой задачи в метод решения, всегда представляют собой значения переменных, то для уменьшения количества уровней и сложности семантических описаний задач целесообразно обобщить тип фактически передаваемого параметра от значения переменной до значения выражения. При этом значением параметра, передаваемого в вызываемую подзадачу или метод, могут быть константа, значение переменной или значение выражения с полностью определенными операндами.

Данной ситуации будут соответствовать измененные шаблоны простой и сложной задачи из [2], приведенные на рис. 1 и 2.

```

task:
  declaration: {id: <идентификатор задачи>, name: <название задачи>, a_task: <id абстрактной задачи>}
  input:
    - object: <идентификатор объекта>
      parameters:
        - {param: <объявление параметра>, name: <название параметра>, type: <тип данных>}
  inputPredicate:
    - <условия на входы для выполнения задачи>
  output:
    - object: <идентификатор объекта>
      parameters:
        - {param: <объявление параметра>, name: <название параметра>, type: <тип данных>}
  outputPredicate:
    - <условия на выходы>
  constraints:
    requirements: <список требований (характеристик) к задаче>
    tests:
      - input(<проверочные входы>), output(<проверочные выходы>)
  method:
    id: <идентификатор метода решения задачи>
    parameters:
      input:
        - <список входов метода>
      output:
        - <список выходов метода>
#task END <id>

```

Рис. 1. Шаблон простой задачи

task:

```
declaration: {id: <идентификатор задачи>, name: <название задачи>, a_task: <id абстрактной задачи>}
input:
  - object: <идентификатор объекта>
  parameters:
    - {param: <объявление параметра>, name: <название параметра>, type: <тип данных>}
inputPredicate:
  - <условия на входы для выполнения задачи>
output:
  - object: <идентификатор объекта>
  parameters:
    - {param: <объявление параметра>, name: <название параметра>, type: <тип данных>}
outputPredicate:
  - <условия на выходы>
constraints:
  requirements: <список требований (характеристик) к задаче>
  tests:
    - input(<проверочные входы>), output(<проверочные выходы>)
implementation:
  if(<условие вызова подзадачи>)
  {
    subtask:
      declaration: {id: <идентификатор подзадачи>, name: <название подзадачи>}
      input:
        - <список входов подзадачи>
      output:
        - <список выходов подзадачи>
      #subtask END <id>
  }
  else
  {
    subtask:
      declaration: {id: <идентификатор подзадачи>, name: <название подзадачи>}
      input:
        - <список входов подзадачи>
      output:
        - <список выходов подзадачи>
      #subtask END <id>
  }
  innerOutputs:
    - <список соответствия выходов задачи выходам подзадач>
#task END <id>
```

Рис. 2. Шаблон сложной задачи с использованием условного оператора

В приведенных шаблонах списки соответствия входных и выходных параметров задачи, подзадач и методов задаются в формате:

- {param: <объект>.<объявление параметра>, equalTo: <выражение>}

где <выражение> – любое допустимое в языке программирования выражение, возвращающее одно значение требуемого типа.

С использованием приведенных шаблонов описание задачи нахождения оптимальной цены нового товара будет иметь следующий вид:

1) описание задачи

task:

declaration: {id: 10, name: Оптимальная цена нового товара, a_task: 41.1}

input:

- object: 19.1

parameters:

- {param: k_max, name: максимальный объем продаж аналогичного товара, type: double}
- {param: price_max, name: максимальная цена аналогичного товара, type: double}
- {param: e, name: начальные затраты, type: double}
- {param: cost, name: себестоимость единицы товара, type: double}

inputPredicate:

- (k_max > 0) & (price_max > 0) & (e >= 0) & (cost > 0)

output:

- object: 19.2

parameters:

- {param: price, name: цена единицы товара, type: double}

outputPredicate:

- Null

constraints:

requirements: {performance: t = 0.001 s, memory: < 2 kB, external: Null}

tests:

- input(70000,350,700000,110), output(230)

method: 1001

parameters:

input:

- {param: (1.7).coef_a, equalTo: (19.1).k_max / (19.1).price_max}
- {param: (1.7).coef_b, equalTo: (19.1).k_max + (19.1).cost * (19.1).k_max / (19.1).price_max}
- {param: (1.7).coef_c, equalTo: -((19.1).e + (19.1).cost * (19.1).k_max)}

output:

- {param: (19.2).price, equalTo: ((1.8).x1 + (1.8).x2) / 2}

#taskEND 10

2) описание метода

method:

declaration: {id: 1001, name: Решение квадратного уравнения, a_task: 41.1.3}

input:

- object: 1.7

parameters:

- {param: coef_a, name: Коэффициент при x^2 , type: double}
- {param: coef_b, name: Коэффициент при x^1 , type: double}
- {param: coef_c, name: Коэффициент при x^0 , type: double}

inputPredicate:

- Null

output:

- object: 1.8

parameters:

- {param: x1, name: Корень 1, type: double}
- {param: x2, name: Корень 2, type: double}

outputPredicate:

- Null

constraints:

characteristics: {performance: t = 0.001 s, memory: 1.5 kB, external: Null}

tests:

```

- input(1,3,2), output(-1,-2)
- input(1,-3,-4), output(-1,4)
source:
{
discriminant = coef_b * coef_b - 4 * coef_a * coef_c;
if (discriminant > 0)
{
x1 = (-coef_b + System.Math.Sqrt(discriminant)) / (2 * coef_a);
x2 = (-coef_b - System.Math.Sqrt(discriminant)) / (2 * coef_a);
}
else if (discriminant == 0)
x1 = x2 = (-coef_b + System.Math.Sqrt(discriminant)) / (2 * coef_a);
else
x1 = x2 = -1;
}
}
#methodEND 1001

```

Таким образом, возможность передачи фактических параметров и получения результата в виде значений выражений повышает уровень абстракции и универсальность подзадач и методов решения подзадач, что уменьшает объемы семантических описаний и повышает быстродействие механизма самомодификации АДИС.

Библиографический список

1. Дрождин В.В, Шалаев А.А. Модель самомодификации информационной системы // Международный научный институт «Educatio». 2015. № 7(14), Ч. 2. С. 20-24.
2. Дрождин В.В., Шалаев А.А. Язык разметки программ в самоорганизующейся информационной системе // Проблемы информатики в образовании, управлении, экономике и технике: сб. ст. XIV Междунар. науч.-техн. конф. Пенза: ПДЗ, 2014. С. 96–104.
3. Дрождин В.В., Шалаев А.А. Программная реализация подсистемы самомодификации и самодоохранения адаптивной информационной системы // Проблемы информатики в образовании, управлении, экономике и технике: сб. ст. XV Междунар. науч.-техн. конф. Пенза: ПДЗ, 2015. С. 48–57.

Дрождин Владимир Викторович

Пензенский государственный
университет,
г. Пенза, Россия
E-mail: drozhdin@yandex.ru

Drozhdin V.V.

Penza State University,
Penza, Russia

Шалаев Александр Александрович

Пензенский государственный
университет,
г. Пенза, Россия
E-mail: shell@live.ru

Shalaev A.A.

Penza State University,
Penza, Russia