Decomposition of the GUI by means of modified algorithm algebra

Abstract

Using operations of cyclic sequencing and cyclic paralleling of the modified algebra of algorithms we have built compact model of decomposition of the formulas processing system of algebra of algorithms. We have developed the generalized abstract model of user graphic interface of a computer system. Based on the generalized model we have created the model of a user interface fragment for formulas processing of algorithm algebra.

Keywords: control icon, graphic interface, GUI, icon model, algorithm algebra, algorithm formula processing.

1. Introduction

The study [1] describes, by means of algebra of algorithms [2-5], the mathematical model of the formulas processing system of algebra of algorithms (SOFAL). The developed model has three levels of decomposition that are done according to the criterion of functional purpose of subsystem. At the first level the SOFAL system is divided into subsystems. In particular the system is decomposed into subsystems that are designed to process the operations of sequencing, elimination, paralleling, cyclic sequencing, cyclic elimination, cyclic paralleling and system interface. The model at the second level of the decomposition describes the user system interface, including the menu model [1]. The third level concerns to the submenu and the model of selection of the submenu components [1].

The resulting model of the first level of the system decomposition [1] can be written in more compact form using the operations of cyclic sequencing or cyclic paralleling.

2. Compact Models of the System Decomposition

The use of the operations of the modified algebra of algorithms that is intended to describing the cycles, gives compact formulas of the subsystem decomposition. For example, the use of operations of cyclic sequencing and cyclic paralleling are described by the formulas

\[ \varnothing \left( d_i t_i @P_i : @N_i \right) \]
and

\[ \varnothing( d_i t_i @P_i : @N_i ) \]

where:

\( \varnothing \) – is the character of the operations of cyclic sequencing,
\( \varnothing \) – is the character of the operations of cyclic paralleling,
\( i \) – is the variable of the subsystems.

The general form of the model of the first level of decomposition, according to the criterion of functional purpose, of the system SOFAL (S) in the language of the modified algorithm algebra, using the operation of cyclic paralleling, is described by the formula:

\[ S = \varnothing( d_0 t_0 @P_0 : @N_0 ) \]

Then the general models of the second and third levels of decomposition, according to the criterion of functional purpose, of the system SOFAL are described by the formulas:

\[ S = \varnothing( d_i t_i @P_i : @N_i ) \]

Visual keys of computer system GUI have different parameters. In general, they include the location of the interface, the name, the geometric size, the color, events and so on. Mathematical models that describe the parameters of visual keys of GUI are called the element-feature models.

3. The Element Feature Model of the GUI

To build the element-feature model of the graphic user interface of the computer system we use the modified algebra of algorithms [3-4].

The formula (1) describes the model of decomposition of the element-feature subsystem according to the needed properties.
More compact of the formula can be described by (1a)

\[ E_k = \varepsilon F_k G_k \]

Here: the abstract \( F = E_k \). The \( E_k \) for the SOFAL is expressed by the form [6-11], where:

- \( F_0, F_1, \ldots, F_{24} \) – are implemented by the known parameters (class identifier, system sub-spaces, that are used to connect system resources, local sub-space) and window properties (size, color and so on);
- \( G \) – is the element of the first level nested into the element \( F \).

This is realized by typical element, marked as Grid [6-11], and

\[ \varepsilon G_k = G_0 G_1 G_2 \ldots G_6, \]

where:

- \( G_0, G_1, G_2, \ldots, G_6 \) – are implemented by the known properties of the Grid;
- \( I_0 = K, I_0 = \ast, I_0 = A, I_0 = B \), where \( A \) – are implemented by columns and \( B \) – by lines of the Grid;
- \( I_1 = P \), given by the formula (2).

We can write the formula (2) in a compact form:

\[ I_1 = \varepsilon j_1; \]

where:

- \( j_1 \in 0, 1, \ldots, 11 \),
- \( j_1 = p_0, p_1, p_2, \ldots, p_{11} \),
- \( r_1 \in 0, 1, \ldots, 16 \).

\( I_1 \) – is the abstract which we replace by the known element ToolBarPanel,

\( p_0, p_1, \ldots, p_{11} \) – are implemented by its typical properties,

\( r_0 = \text{Sep} \), \( r_1 = \text{Sep} \), \( r_2 = \text{Sep} \) are implemented by the typical separator,

\( \text{Sep}_0, \text{Sep}_1, \text{Sep}_2 \) and \( \text{Ser}_0, \text{Ser}_1, \text{Ser}_2 \) – are implemented by its known properties;

\( r_3 = \text{Lab} \) – is the abstract which we replace by the known element (class Label).

Lo_0, \ldots, Lo_7 \) and \( Ld_0, \ldots, Ld_3 \) – are implemented by its typical properties,

\( R_{1, i} = \text{But} \), \( R_{1, i} = \text{But} \), \ldots, \( R_{1, i} = \text{But} \), \ldots, \( R_{12, i} = \text{But} \), \( R_{16, i} = \text{But} \) are implemented by the typical button (class Button),

\( B_0, B_1, B_2, B_3, B_4 \) and \( B_5, B_6, B_7 \) with indexes are implemented by typical properties of the button,

\( Im \) – is implemented by the icon on the button, where \( Is, Ic, Ip, Icx, Icx, Icp, Ir, Id, Ik \) and \( Hi \) with indexes, are implemented by the access components to these images;

\( R_{0, i} = \text{Cb} \) and \( R_{0, i} = \text{Cb} \) – are implemented by the typical developing text area (class ComboBox).

In the expanded form \( D \) is given by the formula (3), and in the compact form by (3a)

\[ I_2 = \varepsilon J_2; \]

where:

- \( j_2 \in 0, 1, \ldots, 8 \), and \( J_2 = \varepsilon D_0, D_1, D_2, \ldots, D_8 \),

\( r_2 = 0, I_2 \) – is the abstract which is implemented by the known graphic element DockPanel, and \( D_0, D_1, D_2, \ldots, D_8 \) – are implemented by typical properties;

\( R_0 = M \) – is the menu:

\[ \varepsilon D_0, D_1, D_2, \ldots, D_8. \]

The selection of the menu option \((M_0, M_1, \ldots, M_7)\) makes this formula of multi-elimination:

\[ \varepsilon M_0, M_1, \ldots, M_7. \]
– the components of the menu (“File” (Mf) are the submenu
“New” (Mn), “Open” (Mv), “Record” (Mz), “Record as” (Mzj), “Finish” (Mk);

\[
\begin{array}{c|c|c|c|c|c|c}
D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 \\
M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 \\
\end{array}
\]

– the options (Mo) contain the submenu “Sequencing” (Ms),
“Elimination” (Me), “Paralleling” (Mp), “Cyclic sequencing” (Mc), “Cyclic elimination” (Mce), “Cyclic paralleling” (Mcp);
– the actions (Md) contain the the submenu “Change” (Mz),
“Delete” (Mv), “Properties” (Mw), “Export/Import” (Mb);
– settings (Mn) contain the the submenu “Font size” (Mk), “Database of algorithms” (Ma);
– the application (Mp) contains the submenu “Generate the code” (Mg);
– the help (Mh) contains the submenu “Information” (Mf);
\[\overline{q_0, q_1, q_2, q_3, q_4, q_5} \] – are the components of the submenu (new, open, record, record as, ...).

The names of the submenu are implemented by typical properties of the submenu.

4. Conclusions

By means of the modified algebra of algorithms we have built a compact element-feature model of the first level of the system decomposition of formulas processing of algebra of algorithms. The model contains 31 subsystems. Using the modified algebra of algorithms we have developed the generalized GUI of computer systems. Based on the designed generalized element-feature model we have created the element-feature model of GUI fragment by the using the formulas processing system of algorithm algebra.

5. References


Received: 01.06.2015     Paper reviewed     Accepted: 02.07.2015

Prof. Volodymyr OVSYAK, PhD, eng.

He specializes in theoretical and applied computer science, theory of algorithms, programming, information systems, mathematical modeling of systems, computer simulation and mathematical modeling. Currently he works as an professor in Kielce University of Technology.

e-mail: ovsyak@rambler.ru

Oleksandr OVSYAK, PhD, eng.

He specializes in theoretical and applied computer science, theory of algorithms, programming, information systems, mathematical modeling of systems, computer simulation and mathematical modeling. He works as an associate professor in National University of Culture and Arts in Lviv, Ukraine.

e-mail: ovsjak@ukr.net