Volodymyr OVSYAK [1], Oleksandr OVSYAK [2]
[1] KIELCE UNIVERSITY OF TECHNOLOGY, 7 Tysiaclecia Panstwa Polskiego Ave., 25-314, Kielce, Poland
UKRAINIAN ACADEMY OF PRINTING, 19 Pid Goloskom St., 79-020 Lviv, Ukraine
[2] NATIONAL UNIVERSITY OF CULTURE AND ARTS, 5 Shuchevitcha St., 79-020, Lviv, Ukraine

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

$$\not\subset i\ (d_i\ t_i\ @P_i : @N_i)$$

and

$$\varnothing i(d_i\ t_i\ @P_i : @N_i),$$

where:

$\not\subset$ – 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. In the simplified SOFAL system the variable runs from 0 to 30.

$$i \in \overbrace{0, \overbrace{1, \ldots\ 30}},$$

d_i – is the variable of access specifiers:
t_i – is the variable of the subsystems types,
@ – the identifier of the subsystem,
: – the identifier of inheritance;
N_i – the name of inherited subsystem.
P_i – the name of inheriting subsystem.

$$d_i \in \overbrace{\overbrace{*, \underline{pu}}, \overbrace{\underline{pr}, \underline{re}}, \ldots},$$

where:

* – access key,
$\underline{pu}$ – public,
$\underline{pr}$ – private,
$\underline{re}$ – recommended and other methods of access to the subsystems;

$$t_i \in \overbrace{\overbrace{*, \underline{ab}}, \overbrace{\underline{co}, \underline{pa}}, \ldots}$$

where:

* – access type of the subsystem,
$\underline{ab}$ – abstract,
$\underline{co}$ – constant,
$\underline{pa}$ – partial.

For example:

$$d_0\ t_0\ @P_0 : @N_0{=}\underline{pu}\ \underline{abs}\ @T,$$

$$d_1\ t_1\ @P_1 : @N_1{=}\underline{pu}\ @U{:}T, \ldots ,$$

$$d_{30}\ t_{30}\ @P_{30} : @N_{30}{=}\underline{pu}\ @Gk{:}\underline{Win}.$$

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:

$$S = \varnothing i\ (d_i\ t_i\ @P_i : @N_i).$$

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:

$$S{=}\ \varnothing i\ (d_i\ t_i\ @(P_i = \varnothing j_i Q_{j,i}){:}@N_i),$$

$$S{=}\ \varnothing i\ (d_i\ t_i\ @[P_i = \varnothing j_i \{Qj_{,i} {=}\ \varnothing t_{j,i} F_{t,j,i}\}]{:}@N_i).$$

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.

$$
\begin{aligned}
F = &\Big(\widehat{\widehat{F_0, F_1}, F_2, \ldots, F_{24}} \\
&; \\
&G = \Big(\widehat{\widehat{G_0, G_1}, G_2, \ldots, G_6} \\
&\quad ; \\
&\quad K = \Big(* \; ; \; A \; , \; B \\
&\quad ; \\
&\quad P \\
&\quad , \\
&\quad D \\
&\quad , \\
&\quad S = \Big(\widehat{\widehat{S_0, S_1}, S_2, \ldots, S_{10}} \\
&\qquad ; \\
&\qquad R = \widehat{\widehat{R_0, R_1}, R_2, \ldots, R_{15}} ,
\end{aligned}
\tag{1}
$$

More compact of the formula can be described by (1a)

$$E_k = \widehat{\varnothing e_k E_{e,k};\ G = \varnothing g G_g;\ \widehat{\varnothing i I_i = \varnothing j_i I_{j,i};\ \varnothing r_i R_{r,i}}}. \tag{1a}$$

Here: the abstract $F = E_k$. The E_k for the SOFAL is expressed by the form [6-11];

$$\varnothing e_k E_{e,k} = \widehat{F_0;\ \widehat{F_1;\ \ldots\ , F_{24}}},$$

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

$$\varnothing g G_g = \widehat{G_0,\ \widehat{G_1,\ \widehat{G_2,\ \ldots\ G_6}}},$$

where:

$G_0, G_1, G_2, \ldots G_6$ – are implemented by the known properties of the Grid;

$I_0=K$, $I_{0,0}=*$, $I_{1,0}=A$, $I_{2,0}=B$, 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:

$$I_1 = \widehat{\varnothing j_1 I_{j,1};\ O r_1 R_{r,1}},$$

where:

$$j_1 \in \widehat{0,\ \widehat{1,\ \ldots\ 11}},$$

$$\varnothing j_1 I_{j,1} = \widehat{P_0,\ \widehat{P_1,\ \widehat{P_2,\ \ldots\ P_{11}}}},$$

$$r_1 \in \widehat{0,\ \widehat{1,\ \ldots\ 16}},$$

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

$P_0, P_1, P_2, \ldots, P_{11}$ – are implemented by its typical properties,

$R_{0,1}=Sep$ i $R_{13,1}=Sep$ are implemented by the typical separator,

Sep_0, Sep_1, Sep_2 and Ser_0, Ser_1, Ser_2 – are implemented by its known properties;

$R_{1,1}=Lab$ i $R_{8,1}=Lab$ – are implemented by the typical label (class Label),

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

$R_{2,1}=But$, $R_{3,1}=But$, … $R_{7,1}=But$, $R_{9,1}=But$, … $R_{12,1}=But$, $R_{16,1}=But$ are implemented by the typical button (class Button),

Bs, Be, Bp, Bcs, Bce, Bcp, Br, Bd, Bk, Bi and *Bj* with indexes are implemented by typical properties of the button,

Im – is implemented by the icon on the button, where *Is, Ie, Ip, Ics, Ice, Icp, Ir, Id, Ik* and *Ii* with indexes, are implemented by the access components to these images;

$R_{9,1}=Cb$ and $R_{9,1}= Cb$ – are implemented by the typical developing text area (classs ComboBox).

$$
\begin{aligned}
P = \Big(& \widehat{\widehat{P_0, P_1}, P_2, \ldots, P_{11}} \\
& ; \\
& Sep = \widehat{\widehat{Sep_0, Sep_1}, Sep_2} \\
& ; \\
& Lab = \widehat{\widehat{Lo_0, Lo_1}, Lo_2, \ldots, Lo_7} \\
& ; \\
& But = \widehat{\widehat{Bs_0, Bs_1}, Bs_2 ;\ Im = \widehat{\widehat{Is_0, Is_1}, \widehat{Is_2, Is_3}, Is_4}} \\
& ; \\
& But = \widehat{\widehat{Be_0, Be_1}, \widehat{Be_2, Be_3} ;\ Im = \widehat{\widehat{Ie_0, Ie_1}, \widehat{Ie_2, Ie_3}, Ie_4}} \\
& ; \\
& But = \widehat{\widehat{Bp_0, Bp_1}, \widehat{Bp_2, Bp_3} ;\ Im = \widehat{\widehat{Ip_0, Ip_1}, \widehat{Ip_2, Ip_3}, Ip_4}} \\
& ; \\
& But = \widehat{\widehat{Bcs_0, Bcs_1}, \widehat{Bcs_2, Bcs_3} ;\ Im = \widehat{\widehat{Ics_0, Ics_1}, \widehat{Ics_2, Ics_3}, Ics_4}} \\
& ; \\
& But = \widehat{\widehat{Bce_0, Bce_1}, \widehat{Bce_2, Bce_3} ;\ Im = \widehat{\widehat{Ice_0, Ice_1}, \widehat{Ice_2, Ice_3}, Ice_4}} \\
& ; \\
& But = \widehat{\widehat{Bcp_0, Bcp_1}, \widehat{Bcp_2, Bcp_3} ;\ Im = \widehat{\widehat{Icp_0, Icp_1}, \widehat{Icp_2, Icp_3}, Icp_4}} \\
& ; \\
& Lab = \widehat{\widehat{Ld_0, Ld_1}, \widehat{Ld_2, Ld_3}} \\
& ; \\
& But = \widehat{\widehat{Br_0, Br_1}, \widehat{Br_2, Br_3} ;\ Im = \widehat{\widehat{Ir_0, Ir_1}, \widehat{Ir_2, Ir_3}, Ir_4}} \\
& ; \\
& But = \widehat{\widehat{Bd_0, Bd_1}, \widehat{Bd_2, Bd_3} ;\ Im = \widehat{\widehat{Id_0, Id_1}, \widehat{Id_2, Id_3}, Id_4}} \\
& ; \\
& But = \widehat{\widehat{Bk_0, Bk_1}, \widehat{Bk_2, Bk_3} ;\ Im = \widehat{\widehat{Ik_0, Ik_1}, \widehat{Ik_2, Ik_3}, Ik_4}} \\
& ; \\
& But = \widehat{\widehat{Bi_0, Bi_1}, \widehat{Bi_2, Bi_3} ;\ Im = \widehat{\widehat{Ii_0, Ii_1}, \widehat{Ii_2, Ii_3}, Ii_4}} \\
& ; \\
& Sep = \widehat{\widehat{Ser_0, Ser_1}, Ser_2} \\
& ; \\
& Cb = \widehat{\widehat{Cf_0, Cf_1}, \widehat{Cf_2, Cf_3}} \\
& ; \\
& Cb = \widehat{\widehat{Cs_0, Cs_1}, \widehat{Cs_2, Cs_3}} \\
& ; \\
& But = \widehat{\widehat{Bj_0, Bj_1}, \widehat{Bj_2, Bj_3}}
\end{aligned}
\tag{2}
$$

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

$$I_2 = \widehat{\varnothing j_2 I_{j,2};\ R_{0,2} = \varnothing t_{0,2} M_{t,0,2};\ \varnothing q_{t,0,2} Q_{q,t,0,2}}, \tag{3a}$$

where:

$$j_2 \in \widehat{0,\ \widehat{1,\ \ldots\ 8}},\ \text{and}\ \varnothing j_2 I_{j,2} = \widehat{D_0,\ \widehat{D_1,\ \widehat{D_2,\ \ldots\ D_8}}},$$

$r_2=0$, 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;

$R_{0,2}= M$ – is the menu:

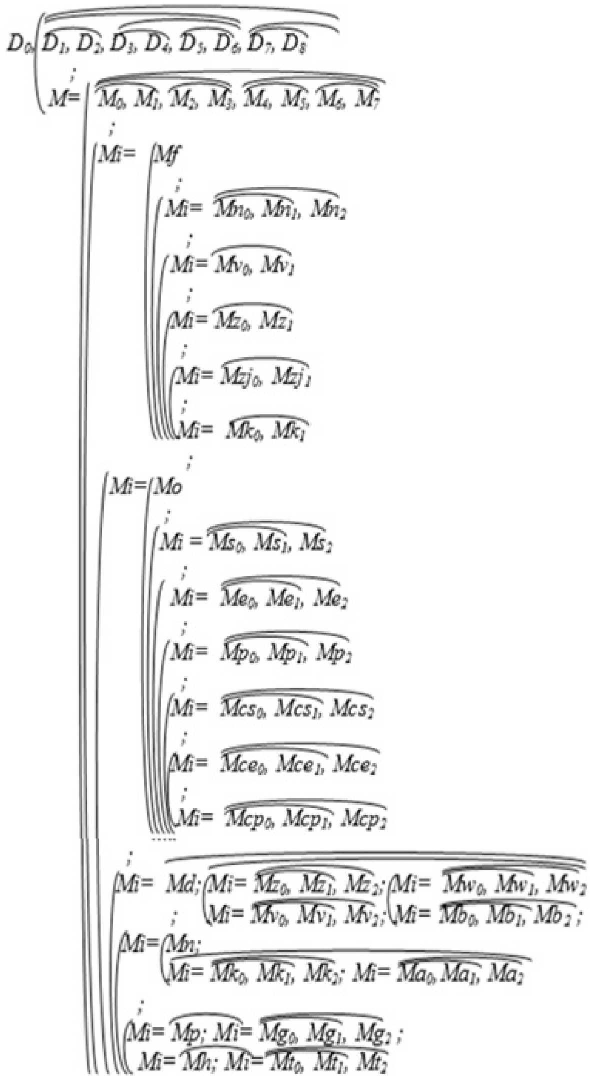
$$\varnothing t_{0,2} M_{t,0,2} = \widehat{M_0,\ \widehat{M_1,\ \ldots\ M_7}},$$

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

$$\overline{M_0;\ M_1;\ \ldots\ M_7;\ *;\ u_{M0};\ u_{M1};\ \ldots\ u_{M7};\ \widehat{\widehat{u_{M0}},\ \widehat{\overline{u_{M1}}},\ \ldots\ \overline{u_{M7}}}},$$

($u_{M0}, u_{M1}, \ldots u_{M7}$ – are the conditions of the selection of the menu options);

– the components of the menu (“File” (*Mf*) are the submenu “New” (*Mn*), ”Open” (*Mv*), ”Record” (*Mz*), ”Record as” (*Mzj*),“Finish” (*Mk*);



(3)

– the options (*Mo*) contain the submenu “Sequencing” (*Ms*), “Elimination” (*Me*), “Paralleling” (*Mp*), “Cyclic sequencing” (*Mcs*), “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” (*Mt*);
$\not\subset q_{t,0,2}Q_{q,t,0,2}$ – 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

[1] Owsiak W., Owsiak A., Owsiak J.: The General Model of the System Decomposition and the Model of the System Interface Fragment. Measurement Automation Monitoring, 2015, nr 05, p. 148 – 150.
[2] Owsiak W., Owsiak A., Owsiak J.: Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych. Wyd. Pol. Opolskiej, Opole, 2005.
[3] Owsiak W., Owsiak A.: Rozszrzerzenie algebry algorytmów. Pomiary, Automatyka Kontrola, vol. 56, No 2, 2010, pp.184 – 188.
[4] Ovsyak O., Ovsyak V.: Modificirovannaja algebra algoritmov i instrumentalnyje sredstwa obrabotki formul algorytmov. Upravlyajuszczije systemy i masziny. No 1, 2013, pp. 27 – 36.
[5] Ovsyak V., Ovsyak O.: Models of layers of information systems. Pomiary Automatyka Kontrola, 2014, nr 06, s. 388 – 390.
[6] Visual Studio 2010. Księga eksperta. Wyd. Helion, 2011.
[7] Visual Studio 2005. Księga eksperta. Wyd. Helion, 2007.
[8] Perry Stephen C. C# i .NET. Wyd. Helion, 2006.
[9] Petzold C.: Programowanie Microsoft Windows w języku C#. Wyd. RM, Warszawa, 2002.
[10] Magennis T.: LINQ to Objects w C# 4.0. Wyd. Helion, Gliwice, 2012.
[11] Brown P.: Silverlight 4 w działaniu. Wyd. APN Promise, Warszawa, 2011.

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