The general model of the system decomposition and the model of the system interface fragment

Abstract

There have been built a fragment of three-level model of the formulas processing system of algebra of algorithms (SOFAL) by means of the modified algorithms algebra. The model of the first level is formed by subsystems. At the second level of decomposition the model is created by graphic and functional parts of subsystems of the first level of decomposition. The third level of decomposition is the model of the system user interface.

Keywords: model of system, processing system of algorithm formulas, sub-system, modified algebra of algorithms, model of the system interface.

1. Introduction

The model of the abstract system, which is used to design the algorithms algebra [1 - 3] is given in the article [4]. The model has the following four levels:

In the first level the model system (S) is divided into the subsystems (P) and is described using cyclic operation of paralleling (Φ), and is described as follows: \( S = \partial P \), where: \( S \) – system; \( P \) – subsystem of the first level of decomposition; \( i \) – variables of first level of decomposition, \( i \in \{0, 1, ..., p - 1\} \); \( p \) – maximum value of variable of first level.

The model of the second layer of subsystem decomposition of the first level is shown in the formula: \( P = \partial Qj \), where: \( Qj \) – subsystem of the second level of decomposition; \( j \) – variables of second level of decomposition, \( j \in \{0, 1, ..., q - 1\} \); \( q \) – maximum value of variable of second level.

Third level of decomposition: \( Qj = \partial F_{ij} \), where: \( F_{ij} \) – subsystem of the third of decomposition; \( t \) – variables of third level of decomposition, \( t \in \{0, 1, ..., s - 1\} \); \( s \) – maximum value of variable of third level.

Fourth layer of partitioning is the following formula of subsystems of algebra of algorithms: \( F_{ij} = \partial H_{tij} \), where: \( H_{tij} \) – subsystem of the fourth of decomposition; \( x \) – variables of fourth level of decomposition, \( x = \{0, 1, ..., h - 1\} \); \( h \) – maximum value of variable of fourth level.

The general formula of the model layers is as follows:

\[ S = \partial P = \partial Qj = \partial H_{tij}. \]

2. The model of the system decomposition of algorithm formulae processing

We use the modified algebra of algorithms to design the mathematical model of the processing system of algorithms formulas (SOFAL) [1 - 3].

The characters of the operations of algebra of algorithms are graphic objects that are recorded over the components. The formulas of algorithms are synthesized from the operations of sequencing, elimination, paralleling and cyclic operations, and can have both horizontal and vertical orientation. The geometrical dimensions of the characters of the operations of algebra of algorithms depend on the size of uniterms and separators. In the process of creating formulas it is necessary to perform such technological operations as cutting, insertion, substitution of formulae fragments and others. In this regard, it is advisable to create a separate system for the synthesis of formulas of algebra of algorithms that will be used in all other systems that deals with components and operations of algebra of algorithms. So we create a general (pu is an identifier of accessibility) and abstract (abs is an identifier of abstraction) subsystem, which is called Term and is denoted by the letter T. The heading of the subsystem is generated by the identifier of access, the identifier of type (abstract, non-abstract), the identifier of the subsystem @, the name of the subsystem and, if there is imitation, by the identifier of imitation (:), and the names of the imitating subsystems. The heading of the subsystem T is written as pu abs @T.

The other subsystems that imitate the subsystem T, will be the subsystems which perform or use the components of processing. First of all, it will be the subsystem of creation of a graphical image of an abstract uniterm (shaded in grey rectangle) with dimensions of 7 to 12 units, which we call Uniterm (denoted by U) with the headline pu @U:T. Beyond the subsystem U the geometric parameters of uniterms are used by the operations of algebra of algorithms. But each of the operations is different, for example, the characters of operations. Therefore we form separate subsystems with the names of operations. The subsystem of processing of the operations of sequencing is called Sequence (S) and we give the heading pu @S:T. Similarly, we form the headings of the subsystems of the processing of elimination operations pu @E:T, paralleling – pu @P:T, cyclic sequencing – pu @CST:T, cyclic elimination – pu @CE:T and cyclic paralleling – pu @CP:T.

To make a link between the processing system of algorithm formulas and operating system Windows we introduce a special subsystem A with the heading pu @A:Appli: where Appli is a shortened name of the subsystem Application of operating system Windows. The subsystem is formed by two parts (part). One of them is used to describe the graphics of the subsystem and it has the heading @A:Appli x: , and the second is for the functional part with the headline pu part @A:Appli. The subsystem is described by sequence of subsystems pu @A:Appli = @A:Appli x: pu part @A:Appli , where x: is referred to the reflection of names x: to the space.

Design of the main window of the user interface and coordination of the subsystem functioning is executed by the subsystem pu @Mf: Win , where Win is the shortened name of the subsystem Window. The subsystem is formed by two subsystems, namely graphic @Mf: Win x: and functional pu part @Mf: Win. Sequence of subsystems makes the formula

\[ pu @Mf: Win = @Mf: Win x: pu part @Mf: Win. \]

To get drop down windows where one can choose the orientation of the operation characters (horizontal, vertical) and separators between uniterms (comma, semicolon) and performance of their setting in the operations we create subsystems of operations:

- sequencing pu @Sf: Win with subsystems @Sf : Win x: and pu part @Sf: Win, which is described as:

  \[ pu @Sf: Win = @Sf: Win x: pu part @Sf: Win; \]

- elimination:

  \[ pu @Ef: Win = @Ef: Win x: pu part @Ef: Win; \]

- paralleling:

  \[ pu @Pf: Win = @Pf: Win x: pu part @Pf: Win; \]
– cyclic sequencing

\[ p_u @ C_f: W_{in} \rightarrow \left[ C_f: W_{in} \right] ; \quad p_u \text{ part } @ C_f: W_{out} \]

– cyclic eliminating:

\[ p_u @ C_f: W_{in} \rightarrow \left[ C_f: W_{in} \right] ; \quad p_u \text{ part } @ C_f: W_{out} \]

– cyclic paralleling:

\[ p_u @ C_f: W_{in} \rightarrow \left[ C_f: W_{in} \right] ; \quad p_u \text{ part } @ C_f: W_{out} \]

To set up the typeface and font size of uniterms of algorithm formulas we create special subsystem \( p_u @ F_f: W_{in} \). The subsystem is formed by graphic and functional subsystems:

\[ p_u @ F_f: W_{in} \rightarrow \left[ F_f: W_{in} \right] ; \quad p_u \text{ part } @ F_f: W_{out} \]

The subsystem

\[ p_u @ D_f: W_{in} \rightarrow \left[ D_f: W_{in} \right] ; \quad p_u \text{ part } @ D_f: W_{out} \]

is introduced for editing formulas of algorithms, which will be done by the replacement of the operation by an empty uniterm or the first or second uniterm operation with two uniters and sequencing area of cyclic operations. The replacement of the selected formula by another or forming the first or the second uniters of the new formula, the operations of sequencing, elimination, paralleling are described in the subsystem

\[ p_u @ D_f: W_{in} \rightarrow \left[ D_f: W_{in} \right] ; \quad p_u \text{ part } @ D_f: W_{out} \]

and one-uniterm of cyclic operations are in the subsystem

\[ p_u @ D_f: W_{in} \rightarrow \left[ D_f: W_{in} \right] ; \quad p_u \text{ part } @ D_f: W_{out} \]

To set the parameters (length, width, coordinates, etc.) of the created objects (forms, panels, buttons, etc.) of user graphical interface we introduce the subsystem

\[ p_u @ F_f: W_{in} \rightarrow \left[ F_f: W_{in} \right] ; \quad p_u \text{ part } @ F_f: W_{out} \]

The subsystem

\[ p_u @ U_g: W_{in} \rightarrow \left[ U_g: W_{in} \right] ; \quad p_u \text{ part } @ U_g: W_{out} \]

is created to design the drop-down window of the task of text uniters of algorithms formulas.

The subsystems \( p_u @ M_c: C_{un} \) and \( p_u @ M_d \) are introduced to initialize and draw the graphic objects of algorithm formulas in the main window. The display in the main window of the created special uniters will be done using the subsystem \( p_u @ S_a \). The algorithms of editing of cyclic operations (replacement of the uniterm or keeping the area) are described in the subsystem

\[ p_u @ D_c: W_{in} \rightarrow \left[ D_c: W_{in} \right] ; \quad p_u \text{ part } @ D_c: W_{out} \]

The subsystem \( p_u @ W_f \) is dedicated to delete a certain fragment or the whole algorithm formula. The subsystem

\[ p_u @ B_a C_f \rightarrow \left[ B_a C_f: W_{in} \right] ; \quad p_u \text{ part } @ B_a C_f: W_{out} \]

is dedicated to create and eliminate algorithms database, and the subsystem

\[ p_u @ B_a \rightarrow \left[ B_a: W_{in} \right] ; \quad p_u \text{ part } @ B_a: W_{out} \]

describes the drop-down window of access to the algorithms database and the record and reading the algorithms database. The subsystem is formed from the algorithm formula of software code of application window.

\[ p_u @ G_k: W_{in} \rightarrow \left[ G_k: W_{in} \right] ; \quad p_u \text{ part } @ G_k: W_{out} \]

The replacement of the formula or its fragment by the uniterm, sequencing, elimination, paralleling, cyclic operations of sequencing, elimination, paralleling are described in the subsystem

\[ p_u @ R_f: W_{in} \rightarrow \left[ R_f: W_{in} \right] ; \quad p_u \text{ part } @ R_f: W_{out} \]

The following formula describes the decomposition of the system into subsystems:

\[ \text{Fig. 1. The fragment of the system model in the form of class diagram} \]

\[ \text{Fig. 2. The fragment of the system model in the form dependence diagram} \]

3. System model fragments in the form of graphs

The built model is implemented on the platform Microsoft Visual Studio .NET. This platform, based on the software code, provides building the models of the system decomposition in the form of graphs. The complete system graph is very bulky and a little informative, so we present only a fragment of the system model represented in the class diagram (Fig. 1) and the dependency diagram (Fig. 2). There is inter-correspondence between algebraic and graph models. For example, in algebraic model the name of the subsystem \( T \) corresponds to the name of the subsystem \( T \) of the graphic model, \( U \) – Uniterm, \( S \) – Sequence, \( E \) – Elimination, \( P \) – Paralleling, \( CS \) – Cyclic Sequence, \( CE \) – Cyclic Elimination, \( CP \) – Cyclic Parallelization, and so on.
As seen from Fig. 1 the subsystem \( T \) is implemented as an abstract class that is followed by all other subsystems – classes of the system graphic model, as illustrated by interrelation with an arrow (see Fig. 2).

The dependency diagram illustrates all available relationships between subsystems – classes. It is obvious that a large number of subsystems – classes and joint links between them make this model difficult to understand and analyze.

4. Model of decomposition of visual-element subsystem

The main subsystem is formed by visual-element (graphic) and functional components:

\[
pu \text{ } @\%M_{F}\text{ } Win = \alpha @\%M_{F}\text{ } Win, \text{ } pu \text{ } part \text{ } @\%M_{F}\text{ } Win.
\]

In this model we use the character % to identify the graphic subsystem which describes the system graphic interface.

The formula \( @\%M_{F}\text{ } Win \) of using the operations of cyclic paralleling can be written as:

\[
@\%M_{F}\text{ } Win = \partial \partial y_{1}\text{-}F_{r},
\]

where: \( r \) – is the variable of the components types, \( r \in \{0, 1, 2, 3, 4\} \), \( \partial y_{1}\text{-}F_{r} \) – the formula which describes the heading of the interface window; \( y_{0} \) – the variable of components of the window heading (icon, name, button of window control), \( y_{1}\text{-}F_{1.4} \) – the window menu; \( y_{1} \) – the variable of the window menu options, \( y_{2} \in \{0, 1, \ldots, 16\} \), \( \partial y_{3}\text{-}F_{2.3} \) – functional elements of the window; \( y_{2} \) – the variable of functional elements of the window, \( y_{3} \in \{0, 1, \ldots, 16\} \), \( \partial y_{4}\text{-}F_{3.4} \) – desktop window, \( y_{4} = 0 \).

The detailed model of decomposition of visual-element subsystem \( @\%M_{F}\text{ } Win(x) \) makes the formula:

\[
@\%M_{F}\text{ } Win(x) = \alpha(T_{i} = \partial S_{i}\text{-}D_{i}, \partial K_{i}\text{-}C_{i}, \partial M_{i}\text{-}T_{i}, M_{i}),
\]

where: \( M_{i} \) – is the formula of the main graphic window of the system means; \( T_{i} \) – is the line of the window title; \( M \) – is the line of the menu; \( K \) – is the line of buttons with certain graphic characters; \( S \) – is scrolling (scrolling element) of the window working area; \( D \) – is the working area for input and editing of algorithm formulas; \( C_{i} \) – are the icons of the main window; \( C_{i} \) – are the names of the system (SOFAL); \( C_{2} \) – are typical icons of rolling up, reducing / enlarging and closing the window, respectively; \( M_{i} \) is the formula: \( M_{i} = \partial m\text{-}File \), where: \( m \) – is the variable and “File” – is the value of the variable \( m \), which is the menu “File”.

The selection \( (\mu_{y_{1}}) \) of the menu option “File” causes to appear the submenu which is described by the formula:

\[
\begin{align*}
\mu_{y_{1}} &= y_{1}\text{-}F_{1.4} = \mu_{y_{1}} = y_{1}\text{-}F_{1.4}, \\
\mu_{y_{1}} &= y_{1}\text{-}F_{1.4} = \mu_{y_{1}} = y_{1}\text{-}F_{1.4}, \\
\mu_{y_{1}} &= y_{1}\text{-}F_{1.4} = \mu_{y_{1}} = y_{1}\text{-}F_{1.4}.
\end{align*}
\]

where \( m_{\text{Nev}}, m_{\text{Open}}, m_{\text{Save}}, m_{\text{SaveAs}}, m_{\text{Exit}} \) – are variables, and their values are “Nev”, “Open”, “Save”, “SaveAs”, “______”, “Exit”, respectively, that are options of the submenu “File”.

The selection of the submenu “Nev”:

\[
C_{\text{Nev}}(y_{1}, y_{2}, y_{3}, y_{4}) = \mu_{y_{1}} = y_{1}\text{-}F_{1.4}.
\]

Similarly it describes the selection of the menu options “Open”, “Save”, “SaveAs”, and “Exit”:

\[
\begin{align*}
C_{\text{Open}}(y_{1}, y_{2}, y_{3}, y_{4}) &= \mu_{y_{1}} = y_{1}\text{-}F_{1.4}, \\
C_{\text{Save}}(y_{1}, y_{2}, y_{3}, y_{4}) &= \mu_{y_{1}} = y_{1}\text{-}F_{1.4}, \\
C_{\text{SaveAs}}(y_{1}, y_{2}, y_{3}, y_{4}) &= \mu_{y_{1}} = y_{1}\text{-}F_{1.4}, \\
C_{\text{Exit}}(y_{1}, y_{2}, y_{3}, y_{4}) &= \mu_{y_{1}} = y_{1}\text{-}F_{1.4}.
\end{align*}
\]

5. Conclusions

The general model of the system SOFAL has been built by means of the modified algebra of algorithms which on the third level is presented as the fragment of interface with the selection of menu elements.

6. References


Received: 15.01.2015 Paper reviewed Accepted: 02.04.2015

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: ovsyak@ukr.net