УДК 004.5

Yevsieiev V.V.
Kharkiv National University of Radioelectronics

VISUAL COMPONENTS FORMAL DESCRIPTION DEVELOPMENT FOR THE AUTOMATED DESIGN OF SOFTWARE PRODUCTS AND MODULES FOR COMPUTER-INTEGRATED PRODUCTION TECHNOLOGICAL PREPARATION SYSTEMS

Within the study on the development of a methodologically justified technology for computer-aided design, the question of a formal representation of forms and their components properties and events description was considered. The systematization of values and linguistic variables representation types was performed depending on their purpose in the main software development environments for production technological preparation computer-integrated systems. An example of basic parameters mathematical and graphical description for form implementations for the RAD Studio XE3 development environment is given.

Key words: *computer-integrated system, production technological preparation, design automation, mathematical description, formal representation.*

Problem Formulation. Effective, cost-effective and competitive work of modern production directly depends on the production technological preparation (PTP) level. At modern stage its increase is reached by introduction of computer-integrated systems for PTP automation. The tasks requiring automation in this direction are difficult to formalize, and as a result, the rules for their automated solution are complexity high level. This leads to high costs for algorithms and software solutions creation. This necessitates the search for alternative solutions, for example, in the creation of new universal models and methods for providing design automation. This will avoid the costs for executing unnecessary and cyclical work in software products and modules design in problems solution for production technological preparation computer-integrated systems (PTP CIS).

Recent researches analysis. Based on literature analysis which addresses issues of software development, namely the consideration of approaches used in management systems for software development [1, p. 5], system approaches to software systems design [2, p. 19], software development technology [3, p. 6], and international standards for software development and operation (ISO/IEC 90003:2014 [4]), life cycle (IEEE Std 1517-2010 [5]), quality indicators (IEEE Std 1061-1998 [6]) we can make a conclusion that existing aspects and recommendations do not allow to use them within an integrated approach for solving the problem of developing unified methodology for software products and modules for PTP CIS design automation field. All considered sources have partial proposals and recommendations on the software products development systematization in general application.

Objective. The main target is to develop visual components formal description, namely forms and their components properties and events, which are the next stage of the study to create a methodologically justified technology for software products and modules for PTP CIS automated design.

Main material statement. To implement a unified methodology for automating the software products and modules for PTP CIS design, the life cycle (LC) model "Jump" was proposed. It made it possible to combine existing concepts and approaches, synthesize them for application in this research (detailed in [7, p. 12]).

After this LC model formalization and using of developed method for synthesizing components and elements of the technical task (TT) structure we obtained form parameters and events set mathematical description $Form_i PE$, as an expression (1) and form structure components set CF_i, represented as (2):

$$Form_i PE \in \underbrace{((mp_1^i, mp_2^i, ..., mp_n^i) \in MP_i}_{\text{Parameters set}} \xrightarrow{\zeta_n} \underbrace{((pp_1^i, pp_2^i, ..., pp_n^i) \in PP_i)}_{\text{Values set}} \wedge \underbrace{((me_1^i, me_2^i, ..., me_n^i) \in ME_i}_{\text{Events set}} \xrightarrow{\zeta_n}$$

$$\underbrace{((ea_1^i, ea_2^i, ..., ea_n^i) \in EA_i)}_{\text{"Linguistic names" set}} \xrightarrow{\phi_n} \underbrace{((z_1, z_2, ..., z_n) \in Z)}_{\text{"Solutions\_containers" set}} \quad (1)$$

where MP_i – parameters set mp_n^i, belonging to the i-th form number; PP_i – values set pp_n^i, which can take mp_n^i as pp_n^i, it must be taken into account that to each $mp_n^i \to pp_n^i$, and a lot of variations that can take pp_n^i are stored in set Z; ME_i – admissible events set may be executed by $Form_i$; me_n^i – unique parameters set; EA_i – "linguistic names" set; Z - solutions containers set.

$$CF_i \in (\underbrace{CD_i^n}_{Element} \in [\underbrace{((pc_1^i, pc_2^i, ..., pc_n^i) \in PC_n^i)}_{\text{Element parameters set}} \xrightarrow{\varepsilon_n} \underbrace{((pp_1^i, pp_2^i, ..., pp_n^i) \in PP_n^i}_{\text{Values set}} \wedge \underbrace{((ce_1^i, ce_2^i, ..., ce_n^i) \in CE_n^i)}_{\text{Events set}} \xrightarrow{\varepsilon_m} \underbrace{((ea_1^i, ea_2^i, .., ea_n^i) \in EA_i)}_{\text{"Linguistic names" set}} \xrightarrow{\phi_n} \underbrace{((z_1, z_2, ..., z_n) \in Z)}_{\text{"Solutions\_container" set}}]) \quad (2)$$

where CD_n^i – i-th form visual components set; PC_n^i – ordered set of parameters for describing visual components pc_m^i, PP_n^i – values set for pp_1^i, which can take the parameters of an element, CE_s^n – events set ce_n^i, which can be processed by the visual component.

To develop values and „linguistic variables" formal representation that will correspond to properties and events description inherent in the form and graphical elements, it is necessary to systematize value and „linguistic variables" representation type depending on their task in software development environments for PTP CIS. In modern development environments (Visual Studio, RAD Studio XE3) analysis, the following values representation types were distinguished, they are presented in the Table 1.

Text type is used for parameter assignment representation, which in most cases are informational in the user interface development. A linguistic type is necessary for assigning a unique definition for „linguistic name" EA_i to a particular event that needs to be processed using the „solutions container" Z specified by the developer. „Linguistic name" is unique in the set EA_i. Boolean type allows the developer to select the activity / passivity of the selected parameter. In most cases it is used to set parameter values $mp_n^i \in MP_1$ и $pc_n^i \in PC_n^i$ for description $Form_i$ and elements CD_i^n. An integer type is used to describe values that are specified by the user in pixels, mainly serve to determine dimensions for $Form_i$, as well as the sizes and coordinates of the location CD_i^n at $Form_i$. The integer negative is used to index the number of possible choices for graphical user interface implementations. Basically, this type is used when numbering image, occurs as a parameter value $mp_n^i \in MP_1$ and $pc_n^i \in PC_n^i$. Text phrase (reserved by

Table 1

Values Types Representation PP_i, PP_n^i, EA_i

Values Type Representation	Application	Example
Text *(a,b,c,...z)* *(a,b,c,...я)*	PP_i, PP_n^i	$\overline{pp_1^i \in PP_i = Name\ of\ the\ main\ form}$ (form name for user, set by developer)
Linguistic *(aaab bbcc,...,aabb)*	EA_i	$ea_1^i \in EA_i =$„*AllCloseForm*" (solutions container linguistic name).
Boolean *(true, false)*	PP_i, PP_n^i	$\underline{pp_1^i \in PP_i = false}$ (non-using specified parameter, set by developer).
Integer ((0,12,...,*n*) ((0,12,...,*n*)	PP_i, PP_n^i	$\overline{pp_1^i \in PP_i = 380}$ (length $Form_i$ situation coordinates).
Integer negative (–1)	PP_i, PP_n^i	Used for indexing, with the value (–1) -indexing is absent, if there is an integer type.
Text phrase *(aa,ab,...aabb)* *(reserved by the development environment)*	PP_i, PP_n^i	$\underline{pp_1^i \in PP_i = clBtnFace}$ (background colour definition for $Form_i$ in environment RAD Studio XE3)

the development environment) type - values of the form of words or abbreviations that are strictly fixed in the development environment and to each specific parameter $mp_n^i \in MP_1, pc_n^i \in PC_n^i$ can belong to a specific set of two or more values pp_n^i.

To simplify values representations formalization PP_i, PP_n^i, EA_i it is proposed to group them into two groups: on the basis of forms parameters description MP_1 and elements PC_n^i, and by forms events $me_n^i \in ME_i$ and elements events $ce_n^i \in CE_n^i$: Then, the formal values representation PP_i, PP_n^i for forms and graphic elements will look like this:

– integer values representation:

$$mp_n^i \vee pc_n^i = \begin{cases} a^1, \ if \ \ a_i \le pp_n^i \le [value], \\ a^2, \ if \ \ [value] < pp_n^i \le [value], \\ \dots \\ a^n, \ if \ \ [value] < pp_n^i \le a_j, \end{cases}$$

where $mp_n^i \vee pc_n^i -$ $n-th$ parameter designation for MP_i and PC_n^i respectively; $a^1, a^2, \dots a^n$ – value range identifiers; a_i, a_i – boundary values that: $a_i \to \min; a_j \to \max$; $[value]$ – selected thresholds for the parameter.

– boolean:

$$mp_n^i \vee pc_n^i = \begin{cases} a^1, \ if \ \ pp_1^i = [false], \\ a^2, \ if \ \ pp_2^i = [true]. \end{cases}$$

where $mp_n^i \vee pc_n^i -$ $n-th$ parameter designation for MP_i and PC_n^i respectively; a^1, a^2 – value range identifiers; $[true, false]$ – logical value definition.

– text word or phrase:

$$mp_n^i \vee pc_n^i = \begin{cases} a^1, \ if \ \ pp_1^i = [word, phrase], \\ a^2, \ if \ \ pp_2^i = [word, phrase], \\ \dots \\ a^n, \ if \ \ pp_n^i = [word, phrase]. \end{cases}$$

where $mp_n^i \vee pc_n^i -$ $n-th$ parameter designation for MP_i and PC_n^i respectively; $a^1, a^2, \dots a^n$ – value range identifiers; $[word, phrase]$ – text or a phrase definition.

– integer negative:

$$mp_n^i \vee pc_n^i = \begin{cases} a^1, \ if \ \ a_i = pp_n^i = -1, \\ a^2, \ if \ \ [1] \le pp_n^i \le [value], \\ \dots \\ a^n, \ if \ \ [value] < pp_n^i \le a_j, \end{cases}$$

where $mp_n^i \vee pc_n^i -$ $n-th$ parameter designation for MP_i and PC_n^i respectively;

$a^1, a^2, \dots a^n$ – value range identifiers; a_i, a_i – boundary values that: $a_i = -1; a_j \to \max$; $[value]$ – selected thresholds for the parameter.

Formal forms events representation ME_i and graphic elements CE_n^i:

– linguistic:

$$me_n^i \vee ce_n^i = \begin{cases} a^1, \ if \ \ ea_1^i = [word], \\ a^2, \ if \ \ ea_2^i = [word], \\ \dots \\ a^n, \ if ea_n^i = [word]. \end{cases}$$

where $mp_n^i \vee ce_n^i -$ $n-th$ parameter designation for ME_i and CE_n^i respectively; $a^1, a^2, \dots a^n$ – value range identifiers; $[word]$ –of the solutions container "linguistic name" definition that is unique in EA_i

Let us give an example of a mathematical description and links graphical representation, guided by the above described, Table 1 and an improved model of Constaintan methodology graphical representation.

Let P exists as a projected software product for solving problems for automated control system (ACS) PTP, which consists of two dialog forms. $Form_1$ acts as the main form for designed ACS PTP, which is represented by a set of $Form_1 PE$ which describes a necessary and sufficient parameters set MP_1 and their values PP_1 which are allowed for each parameter $mp_1^1, .., mp_n^1$ (parameters number and names depend on the selected development environment. Each of them corresponds to a certain $pp_1^1, ..., pp_n^1$ - a set of values that are acceptable to every mp_i^1. Consequently, we can write the following:

$$MP_1 \xrightarrow{\zeta_i} PP_1 \text{ as } \underbrace{\begin{cases} mp_1^1(Caption) \xrightarrow{\zeta_1} \\ mp_2^1(ClientHeight) \xrightarrow{\zeta_2} \\ \vdots \\ mp_n^1(Visible) \xrightarrow{\zeta_n} \end{cases}}_{parameter} \underbrace{\begin{cases} pp_1^1(textname) \\ pp_2^1(1, 2, .., 1200\,pix) \\ \vdots \\ pp_n^1(true, false) \end{cases}}_{value} \quad (3)$$

To simplify the presentation $MP_1 \xrightarrow{\zeta_i} PP_1$ we carry belonging notion in terms of describing the values assignment that are specified by the developer depending on customer's requirements. We introduce basic parameters notion mp_i^1, which are filled in necessarily if the development environment can not construct $Form_1$, and non-basic $\overline{mp_i^1}$ in which values presence pp_n^1 can not be specified (is specified as necessary or is generated by the default

by development environment). An example of a mathematical notation is presented in expression (4). This allows us to describe $Form_i PE$, all necessary parameters for graphical display implementations $Form_i$, which can be specified in the developed automated system for the software design for CIS PTP by expression (5):

$$MP_1 \xrightarrow{\begin{cases} mp_1^1 \xrightarrow{\zeta_1 = textname} pp_1^1 \\ mp_2^1 \xrightarrow{\zeta_2 = 640\,pix} pp_2^1 \\ mp_3^1 \xrightarrow{\zeta_3 = \text{bsSizeable}} pp_3^1 \\ \dots \\ mp_n^1 \xrightarrow{\zeta_n = true} pp_n^1 \end{cases}} PP_1 \qquad (4)$$

$$Form_1\left[Form_1 PE(MP_1) \xrightarrow{mp_3^1 = textname, mp_2^1 = 640, mp_{25}^1 = beSizeable, \dots, mp_n^1 = true} PP_1\right] \qquad (5)$$

Let us consider a mathematical description of necessary and sufficient basic parameters for implementatios of an empty (without elements $CD_n^i \in CF_1$ and events ME_n^i) $Form_i$ for the development environment RAD Studio XE3. Taking into account parameters equality $mp_4^1, mp_5^1, mp_8^1, mp_9^1, mp_{23}^1, mp_{24}^1, mp_{46}^1, mp_{50}^1, mp_{58}^1, mp_{59}^1, mp_{66}^1, mp_{68}^1, mp_{70}^1, mp_{13}^1, mp_{32}^1$ и $mp_{27}^1, mp_{48}^1, mp_{49}^1, mp_{57}^1$ within developing method we propose to group parameters by the same value. This will make possible to simplify description mathematical representation up to next:

$$Form_1 PE[(MP_1) \xrightarrow{mp_3^1 = alNone;\ mp_4^1, mp_5^1, mp_8^1, mp_9^1, mp_{23}^1, mp_{24}^1, mp_{46}^1, mp_{50}^1, mp_{58}^1,} \wedge$$
$$\wedge \xrightarrow{mp_{59}^1, mp_{66}^1, mp_{68}^1, mp_{70}^1 = false;\ mp_6^1 = 255;\ mp_{10}^1 = bdLeftToRight;\ mp_{12}^1 = bsSizeable;} \wedge$$
$$\wedge \xrightarrow{mp_{13}^1, mp_{32}^1, mp_{62}^1, mp_{64}^1 = 0;\ mp_{14}^1 = NameFormCaption;\ mp_{15}^1 = 212;\ mp_{16}^1 = 418;} \wedge$$
$$\wedge \xrightarrow{mp_{17}^1 = clBtnFace;\ mp_{19}^1, mp_{27}^1, mp_{48}^1, mp_{49}^1, mp_{57}^1 = true;\ mp_{20}^1 = crDefault;} \wedge$$
$$\wedge \xrightarrow{mp_{22}^1 = dmActiveForm;\ mp_{25}^1 = dkDra;\ mp_{26}^1 = dmManual;\ mp_{29}^1 = fsNormal;} \wedge$$
$$\wedge \xrightarrow{mp_{30}^1 = TGlassFrame;\ mp_{31}^1 = 250;\ mp_{35}^1 = htContext;\dots;\ mp_{44}^1 = Form1;\ mp_{51}^1 = 96;} \wedge$$
$$\wedge \xrightarrow{mp_{53}^1 = pmNone;\ mp_{55}^1 = poDefaultPos\square nl\square;\ mp_{56}^1 = po\square\square oportional\ mp_{60}^1 = 10;} \wedge$$
$$\wedge \xrightarrow{mp_{63}^1 = tipDontCare;\dots;\ mp_{67}^1 = clBlac;\ mp_{71}^1 = 434;\ mp_{73}^1 = \square sNormal} \square PP_1\ )\square$$

Let us consider events mathematical description $(me_1^1,\dots,me_n^1) \in ME_1$ and "linguistic variables" $(ea_1^1,\dots,ea_n^1) \in EA_1$ as well as the "solutions container" $(z_1, z_2, \dots, z_n) \in Z$ inherent in each $Form_i$ and belong to the set $Form_1 PE$.

$$ME_1 \xrightarrow{\begin{cases} me_1^1 \xrightarrow{\P_1 = ea_1 \text{"linguistic variable 1"} = \phi_1} z_1(\text{"solutions container 1"}) \\ me_2^1 \xrightarrow{\P_2 = ea_2 \text{"linguistic variable2"} = \phi_2} z_2(\text{"solutions container 2"}) \\ me_3^1 \xrightarrow{\P_3 = ea_3 \text{"linguistic variable 3"} = \phi_3} z_3(\text{"solutions container 3"}) \\ \dots \\ me_n^1 \xrightarrow{\P_n = ea_n \text{"linguistic variable n"} = \phi_n} z_n(\text{"solutions container n"}) \end{cases}} Z$$

We take into account such an opportunity as the use of one "linguistic variable" ea_n^1 to describe two or more events $(me_1^1,\dots,me_n^1) \in ME_1$ at $Form_1$, each ea_n^1 "linguistic variable" has its own unique $z_i \in Z$ "solutions container", which contains the program code.

$$Form_1 PE[(ME_1) \xrightarrow{me_1^1,\ me_4^1 = \text{"linguistic variable 1"};} \wedge$$
$$\wedge \xrightarrow{me_2^1 = \text{"linguistic variable 2"};} \wedge \qquad (6)$$
$$\wedge \xrightarrow{me_3^1 = \text{"linguistic variable 3"}} (Z)]$$

For parameters description PC_n^i and their values PP_n^i of graphic elements CD_1^n, which are an integral part $Form_1$ it is suggested to use the record type represented by the expression (2), and events interaction description CE_n^i by "linguistic names" set EA_i with "solutions container" Z respectively (6).

Graphic interaction model fragment $Form_1$ and $Form_2$ using a parameters and events value formal description is shown in the Figure 1:

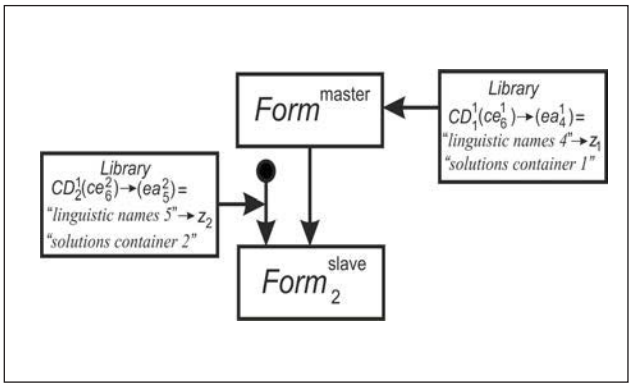

Figure 1. Graphic interaction model fragment $Form_1$ and $Form_2$

Conclusions. Visual components formal description for development environments (RAD Studio XE3, Visual Studio), which is based on form properties and events mathematical description, as well as user's interface elements. A new interaction between the main elements construction graphical representation for objects interaction within software is proposed. In the future, on the basis of the proposed solutions, it is planned to develop a language for software design for CIS PTP. These researches results are implemented in the "Automated System for Software Designing for Corporate Information Systems for PTP «CAD-Programing Code»" certificate of authorship № 74576, 09.11.2017.

References:

1. Бушуев С.Д., Бушуева Н.С. Современные подходы к развитию методологий управления проектами. Зб. наук. пр. Управління проектами та розвиток виробництва. 2005. Вип. №1(13). с. 5-19.
2. Липаев В.В. Проектирование и производство сложных заказных программных продуктов. Москва, 2011. 408 с.
3. Соловьев Н.А., Чернопрудова Е.Н. Системы автоматизации разработки программного обеспечения. Оренбург, 2012. 191 с.
4. ISO / IEC 90003: 2014. Software Engineering. Guidelines for the Application of ISO 9001:2008 to Computer Software. IEEE, 2015. 54 p.
5. IEEE Std 1517-2010. System and Software Life Cycle Processes. Reuse Processes. IEEE, 2010. 51 p.
6. IEEE Std 1061-1998. Standard for a Software Quality Metrics Methodology. IEEE, 2005. 26 p.
7. Невлюдов І.Ш., Євсєєв В.В., Демська А.І. Розробка моделі життєвого циклу розробки програмного продукту та програмних модулів для КІС ТПВ. Научно-технічний журнал «Технологія приладобудування». 2017. №1. С. 12-16.

РАЗРАБОТКА ФОРМАЛЬНОГО ОПИСАНИЯ ВИЗУАЛЬНЫХ КОМПОНЕНТОВ ДЛЯ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ И МОДУЛЕЙ ДЛЯ КОМПЬЮТЕРНО-ИНТЕГРИРОВАННЫХ СИСТЕМ ТЕХНОЛОГИЧЕСКОЙ ПОДГОТОВКИ ПРОИЗВОДСТВА

В рамках исследования по разработке методологически обоснованной технологии автоматизированного проектирования рассмотрен вопрос о формальном представлении описания свойств и событий форм и их компонентов. Проведена систематизация типов представления значений и лингвистических переменных в зависимости от их назначения в основных средах разработки программных продуктов для компьютерно-интегрированных систем технологической подготовки производства. Приведен пример математического и графического описания основных параметров для реализаций формы для среды разработки RAD Studio XE3.

Ключевые слова: *компьютерно-интегрированные системы, технологическая подготовка производства, автоматизация проектирования, математическое описание, формальное представление.*

РОЗРОБКА ФОРМАЛЬНОГО ОПИСУ ВІЗУАЛЬНИХ КОМПОНЕНТІВ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ПРОГРАМНИХ ПРОДУКТІВ ТА МОДУЛІВ ДЛЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМ ТЕХНОЛОГІЧНОЇ ПІДГОТОВКИ ВИРОБНИЦТВА

У рамках дослідження з розробки методологічно обґрунтованої технології автоматизованого проектування розглянуто питання про формальне подання опису властивостей і подій форм та їх компонентів. Проведена систематизація типів представлення значень і лінгвістичних змінних у залежності від їх призначення в основних середовищах розробки програмних продуктів для комп'ютерно-інтегрованих систем технологічної підготовки виробництва. Наведено приклад математичного та графічного опису основних параметрів для реалізацій форми для середовища розробки RAD Studio XE3.

Ключові слова: *комп'ютерно-інтегровані системи, технологічна підготовка виробництва, автоматизація проектування, математичний опис, формальне подання.*