Advances in Dynamical Systems and Applications.
ISSN 0973-5321, Volume 16, Number 2, (2021) pp. 441-455
© Research India Publications
https://www.ripublication.com/adsa.htm

GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive Cyber-Design CPPS Development

Igor Nevliudov[1], Vladyslav Yevsieiev[1], Vyacheslav Lyashenko[2], M. Ayaz Ahmad[3]

[1]Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv National University of Radio Electronics, Ukraine.

[1]Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv National University of Radio Electronics, Ukraine.

[2]Department of Media Systems and Technology, Kharkiv National University of Radio Electronics, Ukraine.

[3]Department of Physics, Faculty of Science, University of Tabuk, Saudi Arabia.

Abstract

This article presents a method for formalizing parameters, GUI elements and Windows Forms events for the development of Cyber-Physical Production Systems based on object-oriented programming languages cybernetic component HMI additive cyber design. The proposed method will allow to automate the HMI CPPS development process by setting a unique "Lingustic Variable" which refers to "Container Solution" containing a piece of program code that must be implemented in accordance with the requirements of the technical specifications. The developed method allows to reduce the time for creating an additive cyber design, which will increase the efficiency of managing new and old CPPS modernization development.

Keywords: Industry 4.0, Industrial Internet of Things, Cyber-Physical Production Systems, Digital Twins, Additive Cyber Design, HMI, GUI.

1. INTRODUCTION

Modern production is not possible without Industrial Internet of Things (IIoT) concepts usage, which allow you to create a Digital Twin of real production and implement them in Cyber-Physical Production Systems (CPPS) as a complex synthesis of physical and cybernetic components into a single information space [1]-[4]. One of the difficult tasks in the CPPS creation is the cybernetic component implementation, which not only displays current information about the parts production technological process, but also carries the functions of monitoring, analysis and making intelligent decisions [5]. Analyzing the work of Uwe Schleinkofer, Jorge F. Arinez from General Motors Company (GMC), Xin Ma, Marco Rodrigues, we can conclude that the development of HMI for additive manufacturing is a complex task that must take into account the multifactorial nature and specifics of equipment, technological process and production in general. and correctly display information at all levels of SCADA, MES, ERP [5]-[11]. As a result, articles by Jeremy A., Achraf Skander, Harley Oliffa, B. Ahmad, Seung Woo Lee & Jai-Kyung Lee consider experimental and theoretical approaches for HMI implementations within Micro Smart Facture (MSF) [12]-[16]. Nikolaj Borisov conducts research on the HMI interface launched on a mobile device and comes to the following conclusions that this approach is the most promising for future interactive systems in the automotive industry for the selected stage of production [17]. During a critical analysis of existing approaches to the HMI development, it can be concluded that at the moment the CPPS cybernetic component development the is based on HMI existing analogs empirical experience and prototyping [18]-[23]. As a result, the task of creating new approaches to the development of the CPPS cybernetic component arises – as the additive cyber design based on GUI elements for object-oriented programming languages development automation.

2. DEVELOPMENT OF GRAPHIC ELEMENTS PROPERTIES AND EVENTS FORMALIZATION METHOD FOR OBJECT-ORIENTED PROGRAMMING LANGUAGES

Additive cyber design – this is an HMI developed on the basis of high-level object-oriented programming languages using GUI elements. As a result, it can be argued that the developed additive cyber CPPS design (P), is a collection of *Windows Form* ($Form_n$) and GUI elements sets the number and connections of which are determined by the technical tasks requirements. Let be P as CPPS additive cyber design cyber component, therefore, multiple visual windows (*Windows Form*) $Form_n \subseteq P$, if each element of the set $Form_n$ is a set of P.

Theorem 1. $Form_n = P$ if and only if simultaneously $Form_n \subseteq P$ and $P \subseteq Form_n$ i.e. $Form_n = P \Leftrightarrow Form_n \subseteq P$ and $P \subseteq Form_n$. We denote the condition $P(Form_n)$ defined in this way: not one element exists $Form_n$, which would satisfy the condition of Theorem 1. For example, $P(Form_n) = \{Form_n \neq P\}$.

In the context of this condition, we can say that all elements $(Form_nPE, CF_n) \in Form_n \neq P$. Based on this, under this condition, it can be determined that the set P does not contain a single common element $(Form_nPE, CF_n) \in Form_n$ denote as the empty set $\varnothing$. In this study, this will mean that the set $Form_n$ has no $\xrightarrow{\Xi}$ in the set P and this set is considered redundant and, therefore, $Form_n = \varnothing$.

To prove the fulfillment of the conditions of union and intersection of the proposed sets included in P, it is necessary that they satisfy the following properties:

- commutability 2 and $2'$;
- associativity 3 and $3'$;
- distribution 4 and $4'$.

Theorem 2. Let $Form_n$, $Form_nPE$ and CF_n are arbitrary sets of properties and parameters that are included in the P set. Then they have the following equalities:

1. $Form_n \cup Form_n = Form_n$;

2. $Form_n \cup Form_nPE = Form_nPE \cup Form_n$;

3. $\begin{array}{l}(Form_n \cup Form_nPE) \cup CF_n = \\ = Form_n \cup (Form_nPE \cup CF_n)\end{array}$; (1)

4. $\begin{array}{l}(Form_n \cup Form_nPE) \cap CF_n = \\ = (Form_n \cap CF_n) \cup (Form_nPE \cap CF_n)\end{array}$;

$1'$. $Form_n \cap Form_n = Form_n$;

$2'$. $Form_n \cap Form_nPE = Form_nPE \cap Form_n$;

$3'$. $\begin{array}{l}(Form_n \cap Form_nPE) \cap CF_n = \\ = Form_n \cap (Form_nPE \cap CF_n)\end{array}$; (2)

$4'$. $\begin{array}{l}(Form_n \cap Form_nPE) \cup CF_n = \\ (Form_n \cup CF_n) \cap (Form_nPE \cup CF_n)\end{array}$.

Proof. Each assertions of these properties follows from the operations definition on sets and Theorem 1. For these assertions, it is necessary and sufficient to prove the 4th property. The rest of the operations properties are proved similarly. To simplify mathematical notation, we denote by the left and right sides of equality 4. Let us show that both conditions of Theorem 2 are satisfied. To do this, we first prove that

$\Psi \subseteq \Theta$. To do this, take any arbitrary element $\psi \in \Psi$. Then and only then ψ simultaneously belongs $Form_n \cup Form_n PE$ and CF_n from condition $\psi \in Form_n \cup Form_n PE$ it follows that accordingly $\psi \in Form_n$ or $\psi \in Form_n PE$. Based on this, if $\psi \in Form_n$,then $\psi \in Form_n \cap CF_n$. If $\psi \in Form_n PE$, then $\psi \in Form_n PE \cap CF_n$. Therefore, in any case $\psi \in Form_n \cap CF_n$. So $\psi \in \Theta$.

Let us prove that the reverse inclusion also holds $\Theta \subseteq \Psi$. Let's take an arbitrary $\theta \in \Theta$, then we can write:

$$\theta \in (Form_n \cap CF_n) \cup (Form_n PE \cap CF_n) \Rightarrow \theta \in Form_i \cap CF_n$$

or

$$\theta \in Form_n PE \cap CF_n .$$

If $\theta \in Form_n \cap CF_n \Rightarrow \theta \in Form_n$ and $\theta \in CF_z \Rightarrow \theta \in$ $Form_i \cup Form_i PE$ and $\theta \in CF_n \Rightarrow \theta \in (Form_n \cup Form_n PE) \cap CF_n = \Psi$ Based on this, if $\theta \in Form_n PE \cap CF_n \Rightarrow \theta \in Form_n PE$ and $\theta \in CF_n \Rightarrow$ $\Rightarrow \theta \in Form_n \cup Form_n PE$ and $\theta \in CF_n \Rightarrow \theta \in (Form_n \cup Form_n PE) \cap$ $\cap CF_n = \Psi$. From here $\theta \in \Psi$.

Theorem 2 implies that $\Psi = \Theta$. The rest of the operations properties on the proposed sets are verified similarly.

We define the complement of the set P through the notation $\overline{P}$ as the difference between the universal set I and set P hence $\overline{P} = \{ Form_n PE{:}Form_n PE \in I$ and $Form_n PE \notin P \}$.

Theorem 3. Let us prove the properties of the set difference $Form_n PE$ and CF_n provided that $\{Form_n PE, CF_n\} \in Form_n$, then they must fulfill the following properties:

1. $Form_n PE \cup \varnothing = Form_n PE$;
2. $Form_n PE \cup I = I$; (3)
3. $\overline{Form_n PE \cup CF_n} = \overline{Form_n PE} \cap \overline{CF_n}$;

$1'$. $Form_n PE \cap \varnothing = \varnothing$;

$2'$. $Form_n PE \cap \mathrm{I} = P$; (4)

$3'$. $\overline{Form_n PE \cap CF_n} = \overline{Form_n PE} \cup \overline{CF_n}$.

Proof. First, it is necessary to prove that 3 and $3'$ fulfill the conditions. We denote Λ as left and Δ as right side of equality. Let us show that $\Lambda \subseteq \Delta$ and $\Delta \subseteq \Lambda$.

For any $\lambda \in \Lambda$ performed as $\lambda \notin Form_n PE \cup CF_z \Rightarrow \lambda \notin Form_n PE$ and $\lambda \notin CF_n \Rightarrow \lambda \in \overline{Form_n PE}$ and $\lambda \in \overline{CF_n} \Rightarrow \lambda \in \Delta$. Consequently, $\Lambda \subseteq \Delta$.

For any $\upsilon \in \Delta$ performed $\vartheta \in \overline{Form_n PE}$ and $\lambda \in \overline{CF_n} \Rightarrow \lambda \notin \overline{Form_n PE}$ and $\lambda \notin CF_n \Rightarrow \lambda \notin Form_n PE \cup CF_n \Rightarrow \lambda \in Form_n PE$. Consequently, $\Delta \subseteq \Lambda$.

To represent the structure of P, it is necessary to formalize parameters and events $Form_n^{master}$, within the framework of these studies, we denote by: the set MP_n – a set of parameters $Form_n^{master}$; ME_n – a set of events $Form_n^{master}$; CF_n – a set of GUI elements that are located on $Form_n^{master}$; CD_x^n – GUI element which is described by the following sets: PC_y^x – parameters and CE_z^x – GUI element events of CD_x^n. By PP_n and PP_t^x denote the set of values that the parameters can take $Form_n^{master}$ and GUI element, and through EA_n and EA_p^x – a set of "Linguistic variables" that refer to a set Z_0 – "Solution containers" containing fragments of program code in accordance with the logical name "Linguistic variable". Based on the proposed formalization which makes it possible to present the structure $Form_n^{slave}$ the form of the structure shown in Figure 1.

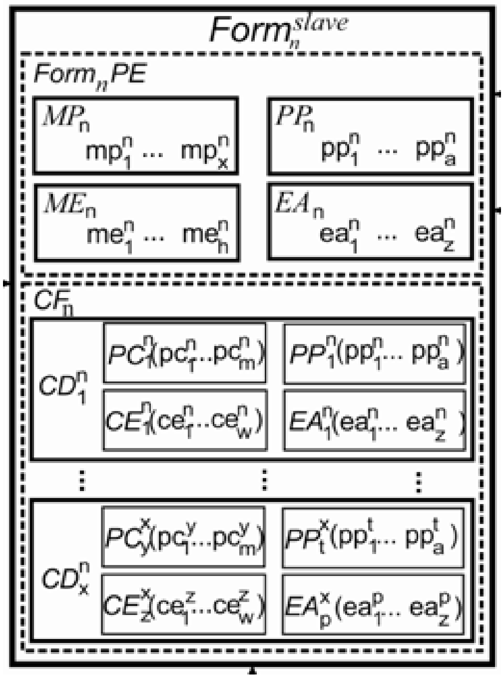

Figure 1: Graphical structure of $Form_n^{slave}$

Based on the proposed formalization of $Form_n^{slave}$, the following generalized mathematical representation is proposed by *Wndows Form* ($Form_n$) as:

$$
\begin{aligned}
& Form_n \in \{Form_n PE \subseteq ((mp_1^n,\ldots,mp_x^n) \in \\
& \in MP_n \cup (pp_1^n,\ldots,pp_a^n) \in PP_n) \wedge \\
& \wedge CF_n \subseteq [((pc_1^y,\ldots,pc_m^y) \in \\
& \in PC_y^1 \cap (ce_1^z,\ldots,ce_w^z) \in CE_z^1) \in CD_1^n] \\
& ,\ldots, \\
& [((pc_1^y,\ldots,pc_m^y) \in PC_y^x \cap (ce_1^z,\ldots,ce_w^z) \in \\
& \in CE_z^x) \in CD_x^n]\}
\end{aligned}
\tag{5}
$$

You can see that record 5 does not describe all the possible parameters that are inherent in $Form_n$, but implies their presence in the form of a parameter sets $mp_1^n,\ldots,mp_x^n$ and many meanings $pp_1^n,\ldots,pp_a^n$ which can take parameters. The set of parameters can vary depending on the development environment and high-level programming language. But, in most cases, the values that they can take are described either in the decimal number system and are measured in pixels (which specify the size of the graphics window, its location and the location of elements inside it, their sizes), or using boolean values ("True","False", which, for example, describe whether an element is displayed or hidden under certain events), as well as with the help of reserved linguistic words that make it possible to simplify the execution of actions with visualization and display of the user interface for the convenience of robots ("Align","BorderStyle","Caption", etc.). Therefore, each parameter can match one or more values pp_a^n (depending on how the values are set).

Based on this, this method will offer two special cases of correspondence.

Special case 1. Define the correspondence ζ between sets MP_n and PP_n, as an arbitrary subset of their products $MP_n \times PP_n$, i.e. $\zeta \subseteq MP_n \times PP_n$. Note that this correspondence consists of ordered pairs. Each pair $(mp_x^n, pp_a^n) \in \zeta$ shows that the parameter $mp_x^n \in MP_n$ matches the value $pp_a^n \in PP_n$ accordance with the ζ. An example of the correspondence for the first particular case between parameter and value is shown in Figure 2.

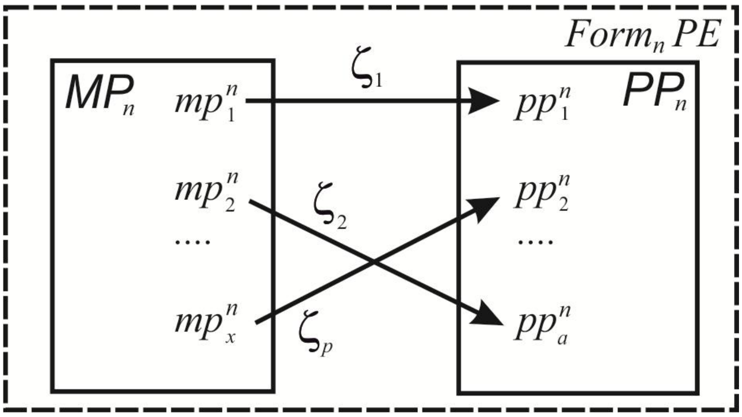

Figure 2: Correspondence of the first special case
(when the parameter mp_x^n only one value belongs to pp_a^n)

Special case 2. The scope of definition of correspondence ζ_1 let's call the set $Dom\zeta = \{mp_x^n \in MP_n$: *that there is a value* pp_a^n, *what* $(mp_x^n, pp_a^n) \in \zeta\}$. Therefore, this definition can be written recursively the set of the match values ζ, which is called the set $\mathrm{Im}\zeta = \{pp_a^n \in PP_n :$ *there is a parameter* $mp_x^n \in MP_n$, *that* $(mp_x^n, pp_a^n) \in \zeta\}$. In Figure 3, we represent the correspondence for the second special case.

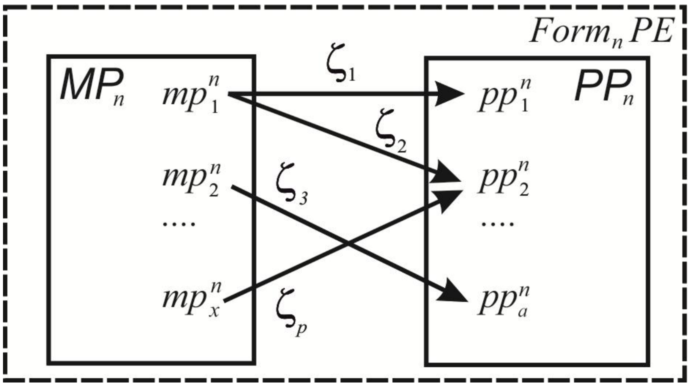

Figure 3: Correspondence of the second special case
(when the parameter mp_x^n can have a set of values pp_a^n).

set of parameters and a set of values $(pp_1^n,..,pp_a^n) \in PP_n$ through the following entry:

- for the first special case

$$mp_1^n \xrightarrow{\zeta} pp_1^n \text{ or } \zeta : (mp_1^n) \in MP_n \to (pp_1^n) \in PP_n; \qquad (6)$$

- for the second special case

$$mp_1^n \xrightarrow{\zeta} (pp_1^n, pp_2^n, pp_{a-1}^n) \tag{7}$$

or

$$\zeta : (mp_1^n) \in MP_n \to (pp_1^n, pp_2^n, pp_{a-1}^n) \in PP_n \ . \tag{8}$$

Let's define the correspondence ζ , as fulfilling such requirements:

- everywhere defined if $Dom\zeta = MP_n$;
- surjective if $\mathrm{Im}\zeta = PP_n$;
- unambiguous if each $mp_x^n \in Dom\zeta$ corresponds to a single parameter pp_a^n from PP_n, that is from $(mp_x^n, pp_a^n) \in \zeta$ and $(mp_x^n, pp_a^n) \in \zeta \Rightarrow pp_a^n = pp_1^n$;
- injective if different parameters $Dom\zeta$ correspond to different parameters of the PP_t, i.e. $(mp_x^n, pp_a^n) \in \zeta$ and $(mp_x^n, pp_a^n) \in \zeta \Rightarrow mp_x^n = mp_1^n$.

If these requirements are met, the mapping will have at least one correspondence (we define it by a partial bijection).

This solution is applicable and sufficient to determine a set of values $(pp_1^n, \ldots, pp_a^n)$, which can take parameters ($(mp_1^n \ldots, mp_x^n) \in MP_n$) to set rendering parameters , that is, all the necessary parameters that make it possible to create an "empty" template *Windows Form* (“window”), and does not take into account events and reactions when interacting with $Form_n$, that is, actions such as the standard built-in events "Close", "Create" and reactions in the form of a "container of decisions", in which the CPPS developer implements options for interacting through the GUI with data, parameters that come from the physical component or solve the HMI functionality tasks on cybernetic level in accordance with the requirement of the technical task and the main goal of the CPPS development. Based on the above, corresponding to 1 and 2 to find the particular case $mp_1^n \xrightarrow{\zeta} pp_1^n$ and $mp_1^n \xrightarrow{\zeta} (pp_1^n, pp_2^n, pp_{a-1}^n)$ definition of $Form_n$ is unsuitable.

To solve them, the sets $(ME_n, EA_n) \in Form_n PE$, as an integral part of the set $Form_n$. Considering the proposed concept of CPPS development process implementations, we will define the purpose of each set.

$(me_1^n, \ldots, me_h^n) \in ME_n$ – set of event parameters, which correspond to a plurality $Form_n PE$, where me_h^n described by a linguistic constant ("Close", "Create", etc.)

and can be extended depending on the development environment within the same object-oriented programming language.

A set of $(ea_1^n,\ldots,ea_z^n) \in EA_n$ introduced to set a natural linguistic description of an event, for example, it can be described in simple words or expressions: "Close a form", "Close a form with saving data", "Close a window with a dialog box", etc. This introduction is required by approval $(mp_1^n \ldots,mp_x^n) \in MP_n \notin (me_1^n,\ldots,me_h^n) \in ME_n$, due to the fact that the elements $pp_a^n \neq ea_z^n$ by the way description values are presented.

As a consequence, there is a need for the existence of a set $(z_1^o, z_2^o,\ldots,z_q^o) \in Z_o$, where Z_o is a set of possible solutions fragments in the form of program code for performing a specific function and procedure that contains a "template" depending on the programming language and development environment. In the future, within the framework of these studies, we will define it as "*Container Solution*", which is determined by setting a unique "*Lingustic Variable*" ea_z^n. Let us give an example of the set interaction $(me_1^n,\ldots,me_h^n) \in ME_n$ through *«Lingustic Variable»* $(ea_1^n,\ldots,ea_z^n) \in EA_n$ and a set $(z_1^o, z_2^o,\ldots,z_q^o) \in Z_o$ – *«Container Solution»* presented in Figure 4.

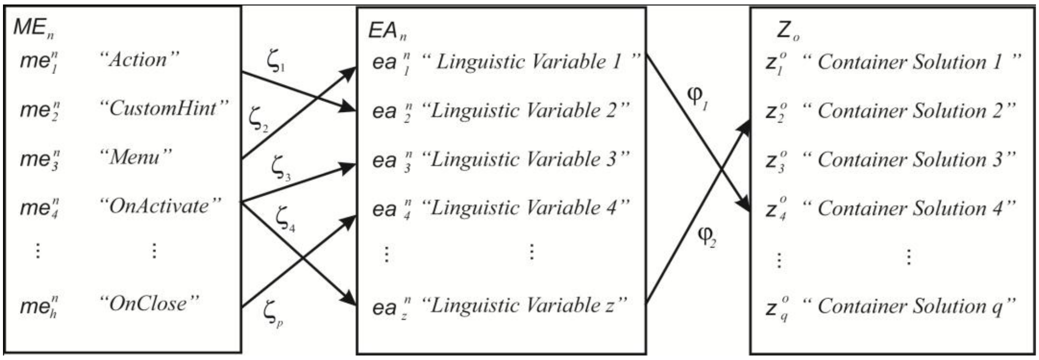

Figure 4: An example of the set interaction $(me_1^n,\ldots,me_h^n) \in ME_n$ using *«Lingustic Variable»* $(ea_1^n,\ldots,ea_z^n) \in EA_n$ and sets of $(z_1^o, z_2^o,\ldots,z_q^o) \in Z_o$ *«Container Solution»*.

We define the existence of an inverse correspondence to the correspondence $\zeta \subseteq ME_n \times EA_n$ as $\zeta' = \{(ea_2^n, me_1^n) : (me_h^n, ea_z^n) \in \zeta\}$. Notice, that $\zeta' \subseteq EA_n \times ME_n$, so ζ' - this is the correspondence between EA_n and ME_n. Figure 5 shows the inverse correspondence to ζ, shown in figure 4.

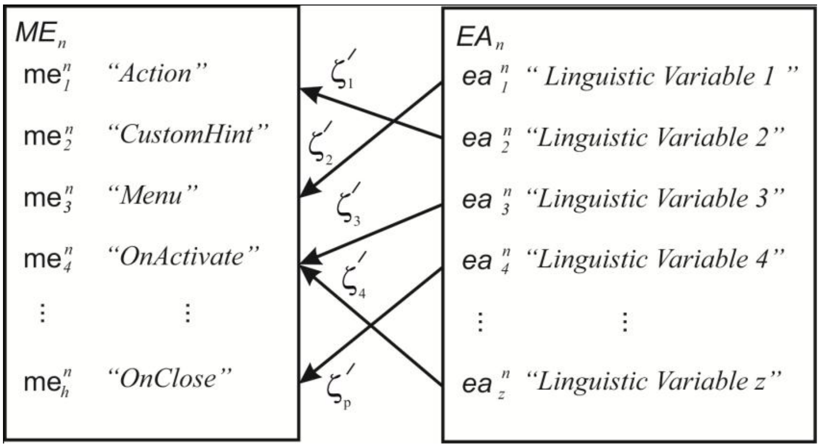

Figure 5: Reverse correspondence to ζ .

In this case, for Figure 5, the mathematical notation will be as follows:

$$\begin{aligned}&\zeta' = \{(ea_1^n, me_3^n), (ea_2^n, me_1^n), (ea_3^n, me_4^n),\\&(ea_4^n, me_h^n), (ea_z^n, me_4^n)\}\end{aligned} \tag{9}$$

To prove the existence of such a solution, it is necessary to determine the compositional correspondence. Therefore, from Figure 4, the compositional correspondence $\zeta \subseteq ME_n \times EA_n$ и $\phi \subseteq EA_n \times Z_o$ is the correspondence $\chi \subseteq ME_n \times Z_o$ such that $\chi = \{(me_h^n, z_q^o):$ *there is an element* $ea_z^n \in EA_n$, *that* $(me_h^n, ea_z^n) \in \zeta$ *and* $(ea_z^n, z_q^o) \in \varphi\}$. Further the compositional correspondence will be written as $\chi = \zeta \circ \varphi$. Then the compositional correspondence for Figure 4 ME_n, EA_n and Z_0 is a set that we represent as a system:

$$\begin{cases}\zeta_2 \circ \varphi_1 = \{(me_{h-1}^n, z_4^0)\}\\ \zeta_4 \circ \varphi_2 = \{(me_h^n, z_2^o)\}\end{cases} \tag{10}$$

Theorem 4. The existence of partial bijections ζ' and $\zeta \circ \varphi$. If $\zeta \subseteq ME_n \times EA_n$ и $\varphi \subseteq EA_n \times Z_o$ possesses the properties of partial bijection, then the following properties must be satisfied:

1. ζ' is a partial bijection between EA_n and ME_n;
2. $\zeta \circ \varphi$ is a partial bijection between ME_n and Z_o.

Proof:

1. Because ζ partially definitely, then ζ' subjective, and therefore ζ surjective then ζ' there is definitely at least one connection everywhere. The two remaining properties are checked in the same way.

2. Because ζ and φ are partially defined everywhere, then their composition are partially defined everywhere, then their composition $\zeta \circ \varphi$ will be defined in at least one parameter me_h^n of ME_n set. As ζ acts on ea_z^n (*«Lingustic Variable»*) EA_n set and φ for all possible z_q^o *«Container Solution»* of set Z_o, then $\zeta \circ \varphi$ is a subjective match. The uniqueness and injectivity are proved similarly.

Let us expand mathematically the set $CF_n \in Form_n$ (in Figure 1) and define it as a set of subsets in Figure 6.

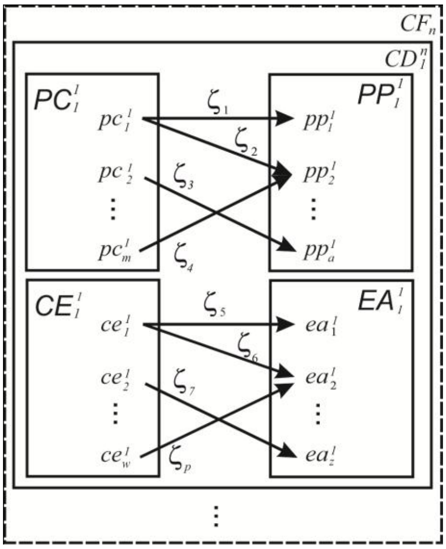

Figure 6: Representation of a set CF_n for describing the properties of controls for working with data.

Let us define the basic properties of the set CF_n as $(CD_1^n,\dots,CD_x^n) \in CF_n$, which contains a certain number of HMI visual graphic elements, necessary and sufficient to implement all functions and operations to achieve the main goal of CPPS development. By visual components we mean GUI interface elements ("Grid", "Table", "Menu", "Button", etc.). Record CD_1^n means: n is a number of $CF_n \in Form_n$, to which the visual component belongs, and 1 is its numbering in the set CF_n. This is due to the fact that many descriptions of visual components

$CD_1^n,\ldots,CD_x^n$, which form the control interaction HMI with data (which, in turn, can have a different purpose, and, therefore, a different set of parameters both for the graphical user presentation and for the events that they can handle). Note that one element CD_x^n can be used many times, but, at the same time, have different purposes in accordance with the logic laid down by the developer. As a result $CD_x^n \in (PC_y^x, PP_t^x, CE_z^x, EA_p^x)$, having the properties proved in Theorem 1-4 and since:

$$\begin{gathered} CF_n \in \{CD_1^n \in (PC_y^1, PP_t^1, CE_z^1, EA_p^1),\ldots \\ \ldots, CD_x^n \in (PC_y^x, PP_t^x, CE_z^x, EA_p^x)\} \end{gathered},$$

It is an integral part of $Form_n \in P$. To simplify the implementation of the method and based on the features of the main reference parameters describing geometric shapes imaging elements take the set $CF_n \in (PP_t^n, EA_p^n) = Form_n(PP_t, EA_p)$ and contains all valid parameters for describing visual properties PP_t^n and "linguistic names" EA_p^n both forms ($Form_n PE$) and components CF_n, which will allow using one knowledge base in the future. The use of this solution allows, on the basis of two proven special cases (Figure 2, 3), to implement the connection between the elements as:

$$\begin{gathered} \varepsilon = \{(pc_1^n, pp_1^n),(pc_1^n, pp_2^n),(pc_2^n, pp_a^n),(pc_m^n, pp_2^n), \\ (ce_1^n, ea_1^n),(ce_1^n, ea_2^n),(ce_2^n, ea_z^n),(ce_w^n, ea_z^n)\} \end{gathered} \tag{11}$$

provided that all properties ζ which are proved above are preserved for ε. As a consequence of this interaction, the set $(ce_1^n,\ldots,ce_w^n) \in CE_n$ through *«Lingustic Variable»* $(ea_1^n,\ldots,ea_z^n) \in EA_n$ and set $(z_1^o, z_2^o,\ldots,z_q^o) \in Z_o$ *«Container Solution»* is proven.

Based on the above presented evidence of the appropriate links existence between: "Properties" → "Parameters"; "Events" → "Linguistic variables" → "Solution container", allows you to solve the problem of development management automation by additive cyber design CPPS, based on a new architectural-logical model to automate the management of the process of creating complex cyber-physical production systems [24], [25].

3. CONCLUSION

During the research, the authors proposed a new method for formalizing parameters, properties and events inherent in GUI elements and Windows Forms for object-oriented programming languages on the basis of which the additive cyber-design CPPS development is provided. Within the framework of the developed method, the possibility of using "Linguistic variables" for describing events of GUI elements and Windows Forms using the natural knowledge language representation was proposed and proved. This allows you to automate the development of the HMI prototype additive cyber design CPPS by implementing the links between "Events" → "Linguistic variables" → "Solution containers". The "solutions Container" fragment contains the program code required to implement the functions and requirements specified in the specifications stored in the database.

To check the correctness of theoretical studies, on the proposed method basis, a system "Automation of CPPS creation management processes" was developed, which made it possible to reduce the development time of an additive cyber design CPPS of the Manufacturing Execution System (MES) enterprise level by 1.3 times in comparison with standard approaches to software development Rapid Application Development (RAD).

REFERENCES

[1] Yongxin Liao, Eduardo de Freitas Rocha Loures, Fernando Deschamps. Industrial Internet of Things: A Systematic Literature Review and Insights, IEEE Internet of Things Journal, Volume 5, Issue 6, pp. 4515-4525, 2018. (DOI: 10.1109/JIOT.2018.2834151).

[2] Rami Matarneh, Svitlana Maksymova, Zhanna Deineko, Vyacheslav Lyashenko. Building Robot Voice Control Training Methodology Using Artificial Neural Net, International Journal of Civil Engineering and Technology, Volume 8, Issue 10, pp. 523-532, 2017.

[3] Oleksandr Kuzomin, Mohammad Ayaz Ahmad, Hryhorii Kots, Vyacheslav Lyashenko, Mariia Tkachenko. Preventing of technogenic risks in the functioning of an industrial enterprise, International Journal of Civil Engineering and Technology, Volume 7, Issue 3, pp. 262-270, 2016.

[4] Kyu Tae Park, Jehun Lee, Hyun-Jung Kim & Sang Do Noh. Digital twin-based cyber physical production system architectural framework for personalized production, The International Journal of Advanced Manufacturing Technology, Volume 106, pp. 1787-1810, 2020. (DOI: 10.1007/s00170-019-04653-7).

[5] Romulo Gonçalves Lins, Paulo Ricardo Marques de Araujo, Marcio Corazzim. In-process machine vision monitoring of tool wear for Cyber-Physical Production Systems, Robotics and Computer-Integrated Manufacturing, Volume 61, 2020. (DOI: https://doi.org/10.1016/j.rcim.2019.101859).

[6] KlausWucherer. HMI, The Window to the Manufacturing and Process Industry, IFAC Proceedings Volumes, Volume 34, Issue 16, pp. 101-108, 2001. (DOI: 10.1016/S1474-6670(17)41508-4).

[7] Uwe Schleinkofer, Kevin Klöpfer, Marco Schneider, Thomas Bauernhansl. Cyber-physical Systems as Part of Frugal Manufacturing Systems, Procedia

CIRP, Volume 81, pp. 264-269, 2019. (DOI: 10.1016/j.procir.2019.03.046).

[8] David S. Icon, Jorge F. Arinez, Micah T. Collins, Zhuming Bi. Modelling of human–machine interaction in equipment design of manufacturing cells, Journal Enterprise Information Systems, Volume 11, Issue 7, pp. 969-987, 2017. (DOI: 10.1080/17517575.2016.1248495).

[9] Orobinskyi P., Petrenko D., Lyashenko V. Novel Approach to Computer-Aided Detection of Lung Nodules of Difficult Location with Use of Multifactorial Models and Deep Neural Networks. In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM) (pp. 1-5). IEEE. (DOI: 10.1109/CADSM.2019.8779340).

[10] Marco Rodrigues, João Paulo Pereira, Pedro Miguel Moreira. The Design of a HMI for Additive Manufacturing, in 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019. (DOI: 10.23919/CISTI.2019.8760932).

[11] Pere Ponsa, Ramon Vilanova, Àlex Pérez, Bojan Andonovski. SCADA design in automation systems, in 3rd International Conference on Human System Interaction, 13-15 May 2010, Rzeszow, Poland. (DOI: 10.1109/HSI.2010.5514492).

[12] Jeremy A. Marvel, Shelly Bagchi, Megan Zimmerman, Brian Antonishek. Towards Effective Interface Designs for Collaborative HRI in Manufacturing: Metrics and Measures, ACM Transactions on Human-Robot Interaction, Volume 9, Number 4, Article No. 25, 2020 (DOI: 10.1145/3385009).

[13] Svitlana Sotnik, Rami Matarneh, Vyacheslav Lyashenko. System model tooling for injection molding. International Journal of Mechanical Engineering and Technology, Volume 8, Issue 9, pp. 378-390, 2017.

[14] M. Ayaz, T. Sinelnikova, S. K. Mustafa, V. Lyashenko. Features of the Construction and Control of the Navigation System of a Mobile Robot, International Journal of Emerging Trends in Engineering Research, Volume 8, Issue 4, pp. 1445-1449, 2020. (DOI: 10.30534/ijeter/2020/82842020).

[15] B. Ahmad, X. Kong, R. Harrison, J. Watermann, A.W. Colombo. Automatic generation of Human Machine Interface screens from component-based reconfigurable virtual manufacturing cell, in IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, 10-13 Nov 2013, Vienna, Austria. (DOI: 10.1109/IECON.2013.6700369).

[16] Seung Woo Lee & Jai-Kyung Lee. Common-HMI for the monitoring of heterogeneous equipment, Journal of Mechanical Science and Technology, Volume 30, pp. 2667–2673, 2016. (DOI: 10.1007/s12206-016-0528-4).

[17] Nikolaj Borisov, Benjamin Weyers & Annette Kluge. Designing a Human Machine Interface for Quality Assurance in Car Manufacturing: An Attempt to Address the “Functionality versus User Experience Contradiction” in Professional Production Environments, Journals Advances in Human-

Computer Interaction, Volume 2018, Article ID 9502692, 2018. (DOI: 10.1155/2018/9502692).

[18] P. Fantini, G. Tavola, M. Taisch, J. Barbosa, P. Leitao, Y. Liu, M. S. Sayed, N. Lohse. Exploring the integration of the human as a flexibility factor in CPS enabled manufacturing environments: Methodology and results, in IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, 23-26 Oct, Florence, Italy, 2016. (DOI: 10.1109/IECON.2016.7793579).

[19] Erwin Normanyo, Francis Husinu, Ofosu Robert Agyare. Developing a Human Machine Interface (HMI) for Industrial Automated Systems , Journal of Emerging Trends in Computing and Information Sciences, Vol. 5, No. 2, pp. 134-144, 2014. ISSN 2079-8407.

[20] Yunbo Zhang, Tsz-Ho Kwok. Design and Interaction Interface using Augmented Reality for Smart Manufacturing, Procedia Manufacturing, Volume 26, pp. 1278-1286, 2018. (DOI: 10.1016/j.promfg.2018. 07.140).

[21] Karthik Ragunathan, Sampurna Ravindranathan, Naveen SSM. Statistical KPIs in HMI panels, in 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 8-10 Oct. 2015, Noida, India. (DOI: 10.1109/ICGCIoT.2015.7380579).

[22] Zakaria Yahouni, Nasser Mebarki. A New HMI Scheduling Model Implemented on a Real Manufacturing Scheduling System, Service Orientation in Holonic and Multi-Agent Manufacturing, Vol. 762, pp. 331-343, 2018. (DOI: 10.1007/978-3-319 -73751-5_25).

[23] Yu An Duan, Hai Tao Song, Yan Ming Sun. The User-Oriented Design and Practice of HMI, Applied Mechanics and Materials, Volumes 58-60, pp. 1819-1823, 2011. (DOI: 10.4028/www.scientific.net/ AMM.58-60.1819).

[24] Igor Nevliudov, Vladyslav Yevsieiev, Svitlana Maksymova, Inna Filippenko. Development of an architectural-logical model to automate the management of the process of creating complex cyber-physical industrial systems, Eastern-European Journal of Enterprise Technologies, Volume 4, Issue 3(106), pp. 44-52, 2020. (DOI: 10.15587/1729-4061. 2020.210761).

[25] Igor Nevliudov, Vladyslav Yevsieiev, Jalal Hasan Baker, M. Ayaz Ahmad, Vyacheslav Lyashenko. Development of a cyber design modeling declarative Language for cyber physical production systems, J. Math. Comput. Sci., Volume 11, Issue 1, pp. 520-542, 2021. (DOI: 10.28919/jmcs/5152).