



- (51) **International Patent Classification:**
G06F 9/445 (2006.01) *G06F 17/40* (2006.01)
- (21) **International Application Number:**
PCT/RU2013/000505
- (22) **International Filing Date:**
18 June 2013 (18.06.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
2013119434 26 April 2013 (26.04.2013) RU
- (71) **Applicant:** **LIMITED LIABILITY COMPANY "1C"**
[RU/RU]; Pokrovka str., 14/2, bld. 1, room VI, Moscow,
101000 (RU).
- (72) **Inventors:** **PAVLENKO, Denis Anatolyevich;**
Tarkhanskaya str., home 6, apartment 52, Moscow, 109156
(RU). **DERUT, Oday Halikovich;** Molodtsova str., home
4, apartment 129, Moscow, 127221 (RU).
- (74) **Agent:** **KHOSHOONIY, Nataliya Aleksandrovna;** OOO
"1C" Law Group, Dostoevskogo str., 1/21, bld. 1, Moscow,
127473 (RU).

- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) **Title:** METHOD OF AUTOMATED INTERFACE CHANGE (VERSIONS)

(57) **Abstract:** This invention relates to the information technology, especially to the method of automated interface change. It enhances the set of technological tools and allows setting up the user interface in a flexible manner at the stage of implementation and execution. This technological result is achieved by applying the automated interface change method that comprises the following stages: creating a separate system element preliminarily at the development stage, specifying its composing dependent elements and saving such separate element in the storing device; forming a special element to store values of the separate system element; setting the value of the said special system element at the stage of application implementation or execution; downloading that separate system element at the stage of the application execution; obtaining the value of the separate system element from the special value storage element; reconfiguring user interface elements connected with composing dependent elements, by using the obtained value of the separate system element.



METHOD OF AUTOMATED INTERFACE CHANGE (VERSIONS)

Field of the Invention

This invention relates to the information technology, and more specifically to the method of automated interface change.

Background of Invention

The process of automated interface change requires a flexible and user-friendly way to configure the user interface at the stage of implementation or execution of business applications on electronic computers, such as personal computers, laptops, etc.

“Implementation” will hereinafter refer to a process of deploying the software on electronic computers and configuring it subject to a specific environment in order to be run by users.

There are different ways to cope with the said challenge.

Patent Application US No. 2011252397 (publ. October 13, 2011) offers a method of automated interface change that comprises receiving a request from a user; obtaining all system configurations capable of providing the requested service from a dynamic feature model; obtaining a corresponding configuration; reconfiguring resources of the system on the basis of the obtained configuration; and providing the requested service. As a result, the said method makes it possible to provide services optimized for an environment that varies in real time without user interference.

However this method is of limited use as obtaining all system configurations to choose a corresponding configuration and reconfigure system resources on the

basis of the obtained configuration lengthens the process of changing the user interface and requires significant resources to store all possible configurations.

Summary of the Invention

Thus, there is a need to enhance the set of technological tools by developing a rapid and universal method of automated interface change which would enable the user to configure it up by him/herself in a flexible and user friendly way when implementing or executing the application.

To achieve this goal and the specified technical result, the first invention embodiment offers an automated interface change method that comprises the following phases: creating a separate system element preliminarily at the development stage, specifying its composing dependent elements and saving such separate element in the storing device; forming a special element to store values of the separate system element; setting the value of the said special system element at the stage of application implementation or execution; downloading that separate system element at the stage of the application execution; obtaining the value of the separate system element from the special value storage element; reconfiguring user interface elements connected with composing dependent elements, by using the obtained value of the separate system element.

The following invention embodiment makes it possible to achieve the specified technical result by applying the automated interface change method that comprises the following phases: creating a separate system element preliminarily at the development stage, specifying its composing dependent elements and saving such separate element in the storing device; creating an element parameterizing values of the separate system element; forming an element intended to store matches of parameter values with values of the separate system element; entering matches of parameter values with values of the separate system element at the stage of application implementation or execution; setting the parameter values at the stage

of application execution; selecting the value of the separate system element corresponding to the specified parameter value from the element storing matches of parameter values with values of the separate system element; reconfiguring user interface elements connected with composing dependent elements by using the obtained value of the separate system element.

The specific attribute of both embodiments is that where the system element depends on several separate elements, parts of the interface connected with such system element will be displayed if it is permitted at least by one of those separate elements.

Brief Description of the Drawings

Fig. 1 is a logic block diagram illustrating the method of automated interface change according to the first version of this invention.

Fig. 2 is a logic block diagram illustrating the method of automated interface change according to the second version of this invention.

Detailed Description of the Invention

This invention can be implemented on electronic computers (e.g. personal computers, etc.) in a computing system (hereinafter the 'system') designed for overall automation of processes related to economic and organizational activities, various types of accounting, financial and economic analyses in an enterprise in any area of business. The system existing as a set of invariable files comprises a combination of various software tools applied both by developers and users. The system is logically divided into two parts interacting with each other. They are a configuration, i. e. the ability to configure the system subject to the environment of any particular enterprise and tasks to be performed, and a platform to manage the configuration operations. The implementation of this invention requires a relevant infobase to store a set of data used to handle different tasks.

The method of automated interface change enables rapid configuration of the user interface during the implementation or system operations while providing flexible and user-friendly capabilities of setting up the interface by the user him/herself and capabilities of setting up the interface based on parameters of separate elements created in the system.

The method presented in this invention is meant for activating/deactivating when implementing entire functional blocks without changing the application itself, i.e. without deleting or modifying any parts of the system software code. For instance, the developer can single out a capability of working with additional object properties, e.g. the “Goods” properties, as a separate system element. In this case, if the developer disables this capability, all capabilities related to additional “Goods” properties will be hidden in the system interface. With specified settings, the system automatically sets up a new state and makes the interface easier-to-understand, free from unnecessary elements and convenient to work with.

The method presented in the first version comprises creating a separate system element preliminarily at the development stage (Fig. 1-120), specifying its composing dependent elements (Fig. 1-110), and saving such separate element in the storing device. It is also necessary to specify in this element which other system elements it influences. Different system elements, such as attributes, commands can be part of separate system elements. Attributes are catalogue or document fields that are displayed on the form for the user to read or complete. The command is a configuration object used by the developer to describe actions to be performed by the user.

Then a special element is formed (Fig. 1-130) to store values of the separate system element (Fig. 1-120). This special element (Fig. 1-130) has a required property to be specified – the Storage property, for which you can select an object to serve as a source of element values. You can select, for example, such objects as constants, catalogue attributes, and information register resources. Constants store values that change rarely, for example, Organization Name, Tax Payer Number, Director Name, etc. Catalogues are tools used to work with a list of homogeneous

elements of data. Any catalogue has two required attributes – “Code” and “Name” – that are created automatically. Other catalogue attributes that may contain some other information in addition to the name should be determined for each specific catalogue being created. Information registers are intended to store information deployed by a combination of dimensions. Register resources are quantitative or integral data stored in the register. To manage the interface, only those separate elements whose values are stored in Boolean-type attributes can be used.

At the implementation or execution stage (Fig. 1-200), the application sets up a value of the said special system element (Fig. 1-210). At the stage of the application execution (Fig. 1-300), the separate element created in the system is downloaded (Fig. 1-120), the separate element value is obtained from the special value storage element (Fig. 1-130), and user interface elements (Fig. 1-310) corresponding to composing dependent elements (Fig. 1-110) are reconfigured based on the obtained value of the special element (Fig. 1-210). In a particular case, the user interface element (Fig. 1-310) that depends on several separate elements or is part of several separate elements (Fig. 1-120) will be displayed if it is permitted at least by one of those separate elements (Fig. 1-120).

The first version of this invention can be exemplified by a task when it is required, depending on the environment of a particular implementation, to provide for disabling the capability of customer accounting by divisions serving such customers, so that “the Division” field is not displayed in the document form when the document is handled. To this end, a separate element (Fig. 1-120) – “the Accounting by Division” element – should be created. A dependent system element (Fig. 1-110) (e.g. the “Client Receipt” document attribute) should be specified in the separate element composition. Then a special element (Fig. 1-130), “the Accounting by Division” constant, should be formed to store sources of Boolean-type system element values. The special element value (Fig. 1-210) should be set to “False” at the implementation or execution stage (Fig. 1-200 and Fig. 1-300), this disables “the Accounting by Division” separate element (Fig. 1-

120) and the system automatically reconfigures the representation of all interface elements (Fig. 1-310) composing the separate element (Fig. 1-120), such as, for example, fields, commands, report elements, in this particular case the user interface hides “the Division” field.

As regards the second version of this invention, the separate element value (Fig. 2-120) depends on some parameters. To implement this dependency, special elements parameterizing values of separate system elements are used.

This method comprises creating a separate system element (Fig. 2-120) preliminarily at the development stage (Fig. 2-100), specifying its composing dependent elements (Fig. 2-110), and saving such separate element in the storing device. Then the element parameterizing values (Fig. 2-140) of the separate system element (Fig. 2-120) is created. That element should specify a set of objects whose values will determine the selection of separate element values. The list of available objects includes catalogues and dimensions of the information register. In this case, register dimensions show what view points should be used to store the information. One catalogue from the overall list of catalogues and one dimension from each information register can be selected per each parameter value in this list. For example, if a separate element (Fig. 2-120) is stored in the catalogue attribute, then the parameter should contain a link to a particular catalogue element. If the separate element (Fig. 2-120) is stored in the resource of the information register, values of all register dimensions should be specified. In this case, every dimension should be characterized by its parameter. Parameters make it possible to create separate system elements with a result that has not been set just once at the implementation stage but can vary based on data stored in the infobase, based on the parameter of the separate system element (Fig. 2-120).

The next steps include forming an element intended to store matches of parameter values with values of the separate system element (Fig. 2-130) and entering matches of parameter values with values of the separate system elements (Fig. 2-210) at the stage of application implementation or execution (Fig. 2-200).

Then the parameter value (Fig. 2-310) at the stage of the application execution (Fig. 2-300) should be set. The value of the separate element (Fig. 2-320) corresponding to the specified parameter value (fig. 2-310) is selected from the element storing matches of parameter values with values of the separate system element (Fig. 2-130). Then user interface elements (Fig. 2-330) connected with composing dependent elements (Fig. 2-110) is reconfigured based on the separate system element value (Fig. 2-120). If not all parameters are specified for a separate element (Fig. 2-120) of the Boolean type, then “the OR” operation for all values with unspecified parameters is performed. For example, if a separate element (Fig. 2-120) is stored in the information register with “the Organization” dimension and “ the Warehouse” dimension, and only “the Organization” dimension is specified, then the value of the separate element (Fig. 2-120) will be “True” provided that the separate element value (Fig. 2-120) for at least one of warehouses listed in “the Warehouse” dimension is “True”.

The next version of this invention can be exemplified by a task when it is required to display or hide “the VAT Rate” field in documents depending on the organization for which the goods receipt is registered. If the organization maintains separate tax accounting, “the VAT Rate” field should be displayed to enable the user to choose a common (default) or any other (c.g. reduced) rate. If the organization does not maintain separate accounting, this field should be hidden as a default “VAT rate” will be used. To achieve this goal, the method requires creating a separate element – i.e. “the Separate VAT Accounting” element – in the system (Fig. 2-120), specifying its composing dependent elements (Fig. 2-110), and saving the created element in the storing device. The next step is to create the “Organization” element parameterizing (Fig. 2-140) values of the separate element (Fig. 2-120), and form the element for storing matches of parameter values with values of the separate element (Fig. 2-130). Thus the value of the separate element (Fig. 2-120) will be stored in “the Catalogue.Organizations.Attribute.Separate VAT Accounting” catalogue attribute of the Boolean type. At the stage of the application implementation or execution (Fig. 2-200 and Fig. 2-300), “the

Organization” parameter value is set up (Fig. 2-310), and the corresponding “False” or “True” value is selected. Then, if the organization does not maintain a separate tax accounting, “the VAT Rate” field will be hidden, but if it does, this field will be displayed. Similar to the first embodiment example, in a particular case, the user interface element that depends on several separate elements or is part of several separate elements will be displayed if it is permitted at least by one of those separate elements (Fig. 2-120).

Thus, methods of automated interface change offered by this invention embodiments enhances the set of technological tools and allows setting up the user interface in a flexible manner at the stage of implementation and execution.

Claims

1. A method of automated user interface change that comprises the following stages:

- creating a separate system element preliminarily at the development stage, specifying its composing dependent elements and saving such separate element in the storing device;
- forming a special element to store values of the separate system element;
- setting the value of the said special system element at the stage of application implementation or execution;
- downloading that separate system element at the stage of the application execution;
- obtaining the value of the separate system element from the special value storage element;
- reconfiguring user interface elements connected with composing dependent elements by using the obtained value of the separate system element.

2. A method specified in cl. 1, wherein the user interface element depending on several separate elements is displayed if it is permitted at least by one of those separate elements.

3. A method of automated user interface change that comprises the following stages:

- creating a separate system element preliminarily at the development stage, specifying its composing dependent elements and saving such separate element in the storing device;
- creating an element parameterizing values of the separate system element;
- forming an element intended to store matches of parameter values with values of the separate system element;
- entering matches of parameter values with values of the separate system element at the stage of application implementation or execution;

- setting the parameter value at the stage of application execution;
- selecting the value of the separate system element corresponding to the specified parameter value from the element storing matches of parameter values with values of the separate system element;
- reconfiguring user interface elements connected with composing dependent elements by using the obtained value of the separate system element.

4. A method specified in cl. 3, wherein the user interface element depending on several separate elements is displayed if it is permitted at least by one of those separate elements.

1/2

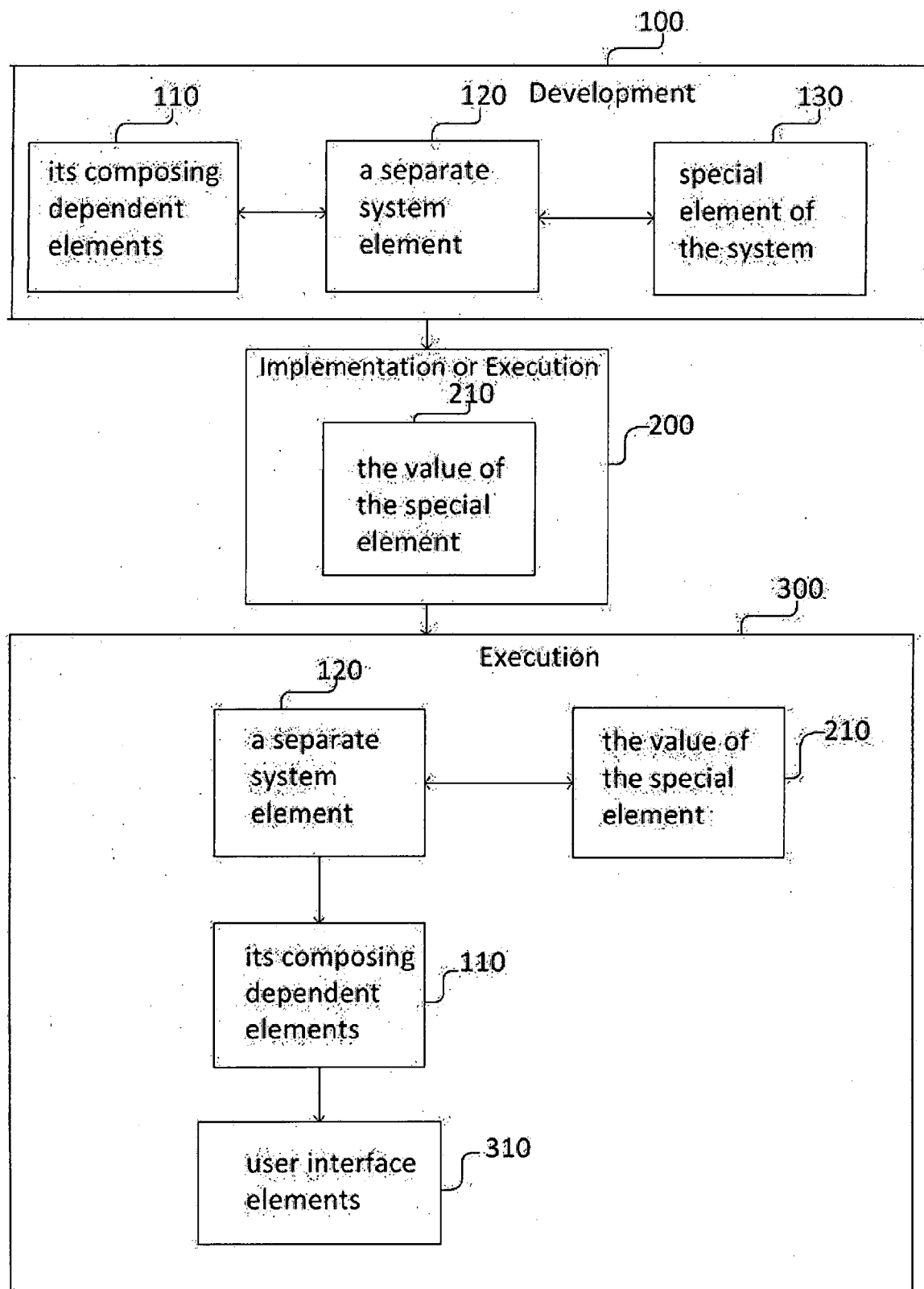


FIG. 1

ЗАМЕНЯЮЩИЙ ЛИСТ

2/2

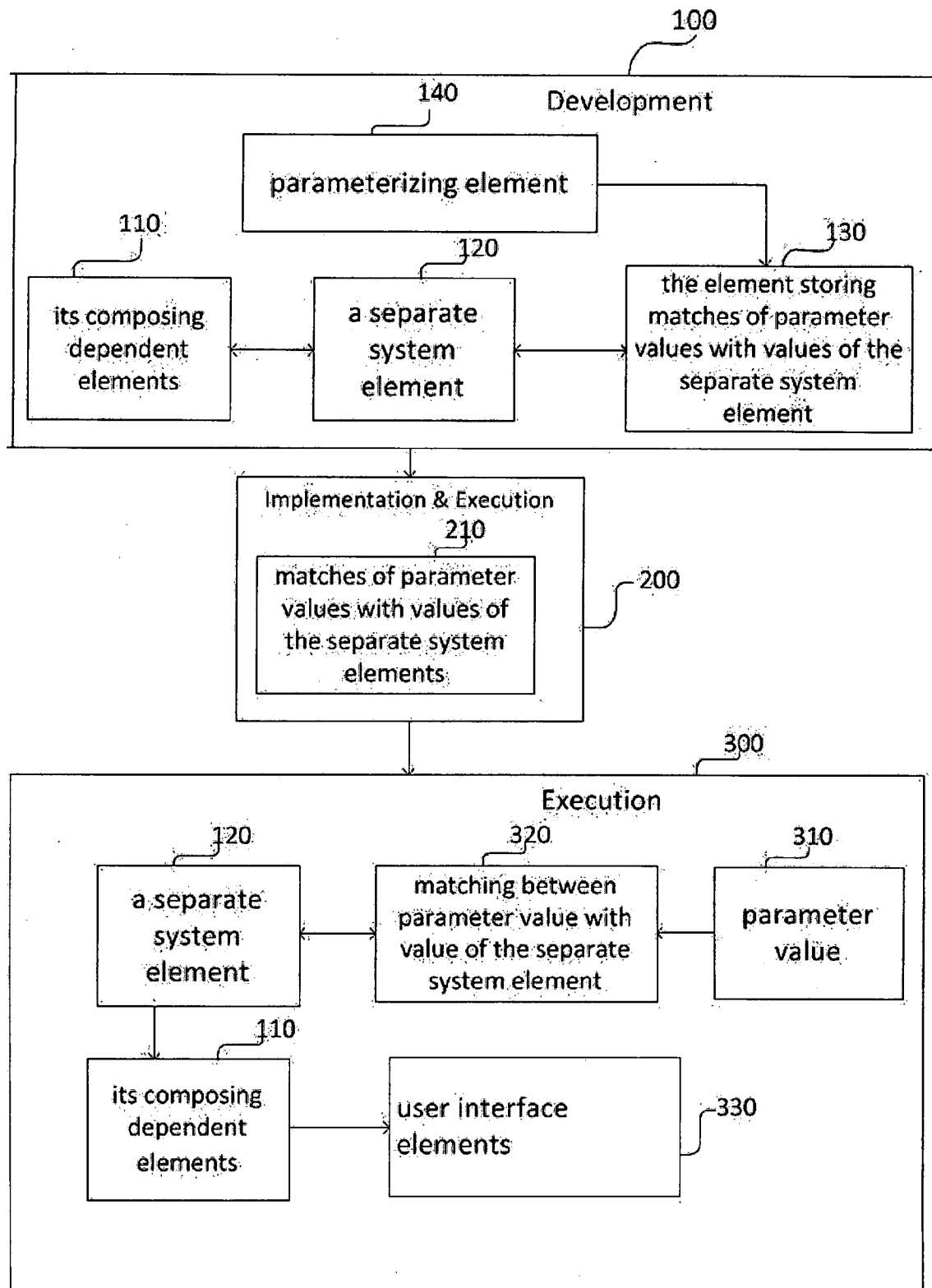


FIG. 2

ЗАМЕНЯЮЩИЙ ЛИСТ

INTERNATIONAL SEARCH REPORT

International application No.

PCT/RU 2013/000505

A. CLASSIFICATION OF SUBJECT MATTER		G06F 9/445 (2006.01) G06F 17/40 (2006.01)	
According to International Patent Classification (IPC) or to both national classification and IPC			
B. FIELDS SEARCHED			
Minimum documentation searched (classification system followed by classification symbols)			
G06F 1/00-1/32, 3/00-3/18, 9/00-9/54, 11/00-11/36, 12/00-12/16, 13/00-13/42, 15/00-15/82, 17/00-17/50, 21/00-21/88, G05B 11/00, 15/00, G06T 1/00-13/00, G09G 5/00-5/42, G06Q 10/00, 50/00, 90/00, H04N 5/00-5/956, 7/00-7/68, 21/00-21/858, H04L 9/00-9/38, 12/00-12/66, 29/00-29/14, H04B 1/00-1/76			
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched			
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)			
DWPI, Espacenet, K-PION, KIPO, PAJ, PatSearch (RUPTO internal), RUPTO			
C. DOCUMENTS CONSIDERED TO BE RELEVANT			
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	
X	US 2006/0136830 A1 (AARON EMERSON MARTLAGE et al.) 22.06.2006, abstract, paragraphs [0009], [0012], [0016], [0022] - [0023], [0028] - [0029], [0034] - [0037], [0041] - [0042], [0044] - [0050], [0060], [0061], claim 9	1-4	
A	WO 2002/059793 A2 (KAUFMAN, MICHAEL, PHILIP et al.) 01.08.2002	1-4	
A	WO 2004/006132 A1 (SOLUTIONS LAB PTE LTD et al.) 15.01.2004	1-4	
A	US 2006/0235548 A1 (THE MATHWORKS, INC.) 19.10.2006	1-4	
A	RU 2365972 C2 (MAYCROSOFT CORPORATION) 27.08.2009	1-4	
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.			
* Special categories of cited documents:		“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family	
“A”	document defining the general state of the art which is not considered to be of particular relevance		
“E”	earlier document but published on or after the international filing date		
“L”	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		
“O”	document referring to an oral disclosure, use, exhibition or other means		
“P”	document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search		Date of mailing of the international search report	
25 December 2013 (25.12.2013)		16 January 2014 (16.01.2014)	
Name and mailing address of the ISA/ FIPS Russia, 123995, Moscow, G-59, GSP-5, Berezhkovskaya nab., 30-1		Authorized officer V. Zhakovich	
Facsimile No. +7 (499) 243-33-37		Telephone No. (495)531-64-81	