UDC 681.5.015 (045)

[1]V. M. Syneglazov,
[2]A. P. Godny

INFORMATION MODEL OF COMPUTER-AIDED DESIGN ENVIRONMENT

National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua, [2]andrewgodny@gmail.com

Abstract*—*The information model of computer-aided design environment is considered. The information model provides the ability to create and develop a design scenario.

Index Terms—Dynamic integration; computer-aided design; database management system.

I. INTRODUCTION

Traditionally, the acronym computer-aided design (CAD) refers to a graphical CAD systems. The graphics systems were automated first. There is a trend of expansion of the meaning of the term CAD. Now this is a set of different software tools, where graphics systems are applied on an equal basis with others. Therefore, the work introduces the term "environment of computer-aided design", that indicates a set of software tools that interact in the process of computer-aided designing [1].

The design process is always the specific actions add-in on pre-defined parameters with given boundary conditions. And this statement is true, both in the initial and the final design stage. Any design process begins with the processing of the project task on the parameters of which is based the initial stage of creation of the object which satisfies the boundary conditions inherited in the job.

Let's consider the process of creating an abstract project. The design assignment is represented as the following components [4]:

– in the form of a table project data, which are the initial parameters, and the boundary conditions data, is specified;

– by drawings and diagrams a graphical description of the location of the application or use of the project are specified;

– by text descriptions subjective and logical characteristics of the project are specified.

II. PROBLEM STATEMENT

At the initial stage the project task design process is performed. The designer selects the priority method for describing the project (graphic, text, table or analytical calculations). Usually the graphical way of describing the project is preferred. This means that in the process of designing the first steps at each stage of the design will be performed on the graphics data of the project. Then depending on the given task the operations on the data of other aspects of the performance are made.

The next stage of design is a direct modification of the design data. Then the change of the project graphic image is held [2]. For example a single object is added. Now you need to specify the correct installation of the new object. For example, if this object is the default, you must specify its dimensions in the reference table and make the appropriate changes on the drawing. Next, you need to make an entry about installing a new object in the explanatory note to the project and specify some of its characteristics [4]. Now you want to bind the new object to existing objects and design task. For this you need to perform some calculations depending on the complexity and purpose of the object, which change the shape and properties of the object or the place of its installation. The calculations are recorded in the explanatory note. Next is the adaptation of the modified object to the terms of the project.

Let's say you want to change the object which is in close relation with other objects. After changing some characteristics of this object designer will need to perform all the above operations for each object associated with the changes to restore the logical connectedness of object. If a project has a set of interconnected objects, the designer practically has to re-perform the design, taking into account all of the binds.

Dynamic data integration of different aspects of the performance is realized during the automated design of control actions on the elements of the environment-aided design [5]. Proposed in the thesis method for dynamic data integration is a method of modeling the interaction of components of the environment of CAD, i. e. a method for dynamic data integration is part of the information model of the environment of CAD.

So it is necessary to create an information model of computer-aided design environment.

III. INFORMATION MODEL STRUCTURE OF COMPUTER-AIDED DESIGN ENVIRONMENT

Information model of environment of computer-aided design, consisting of a control processor (CP), thematic coprocessors and executing processors is suggested (Fig. 1).

The CP controls the behavior of all thematic coprocessors that implement dynamic data integration in the environment of computer-aided design. The thematic coprocessors include: Graphics coprocessor (GP), Table coprocessor (TbP), Text coprocessor (TP) and a Math coprocessor (MP). Thematic coprocessors receive and process commands from the CP and transmit them to the executive processor. Executive processors are software tools that implement the data of one aspect of the presentation. For example, they are: AutoCAD, SolidWorks, dBase, Paradox, MS Word, MathCAD, etc. This information model reproduces real software interaction in a simulated environment of CAD.

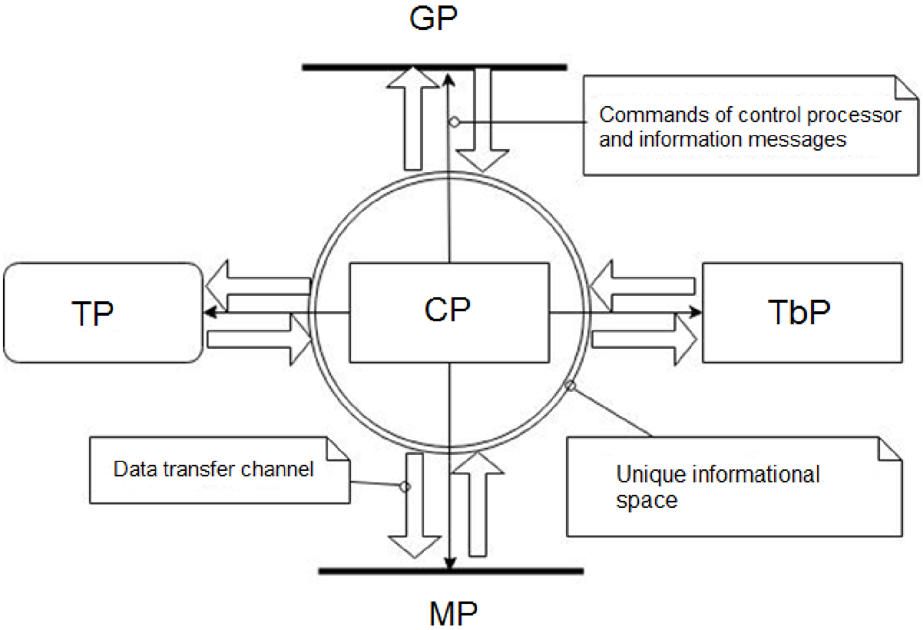

Fig. 1. The scheme of processors control in the information model

Each thematic coprocessor performs certain actions:

– graphics coprocessor automates the procedures of processing the information presented in the form of images and supports appropriate data structures for its storing;

– table coprocessor automates the processing procedures of the information presented in tables and supports the relevant data structures for its storing;

– text coprocessor automates the processing procedures of the information presented in the form of textual descriptions and maintains appropriate data structures for its storing;

– mathematical coprocessor automates the procedure of calculation of the necessary parameters and maintains appropriate data structures for its storing;

– control processor automates the procedures for managing the design process and coordinates the interaction of all thematic coprocessor included in the environment of CAD.

The design process is the execution of certain actions on the data, demonstrating various aspects of the performance. To achieve private design decisions required to perform multiple actions (or commands of CAD). Therefore, the set of commands that are performed to obtain private design solutions is a complex operation. In the design process there are similar complex operations that have the same set of commands and differ only by used data.

Concepts

1. Project is a related description of the design object represented as generic operations performed in the sequence recommended by the design scenario. The project also includes separate commands of thematic coprocessors which are not included in the set of generic operations. The project is carried out manually by the designer using generic operations from the operations library.

2. Generic operation is a set of commands for one or more thematic coprocessors. Coprocessor commands are the procedural component (T) of the design data, connected with each other by K operators. Generalized operation is intended to perform a specific sequence of actions specified by the designer. Generic operations can be connected together using operators for performing connection between different commands from different generic operations.

3. Design scenario is a set of interrelated generic operations recommended for designer. The design scenario specifies the execution sequence of generic operations in the formation of the finished project.

4. The monitor is a tool of CP that monitors the status of the project. The monitor reflects the state of the project as a sequence of thematic processors commands executed. The monitor displays the executed commands and the connection between them and allows editing executed generic operations and overriding the connection between the commands.

IV. Design scenario

This information model provides the ability to create and develop a design scenario that includes the generic operations that are performed in the design process. The result of the generalized operation is the implementation of specific design stage. Generalized operation may consist of commands both from one and from several processors. For example, in the design process you want to open the reference tables, where the data for a specific object is obtained. In this generalized operation the following steps are performed:

– the type of the object is determined;

– the name of the database file containing the directory is determined;

– the parameters for data retrieval keyword is determined;

– the search of the required entries in the database for a specific object is performed.

Using the generalized operation greatly simplifies this process, and now it is not necessary to know for the designer where and how to store the data. Thanks to the generalized operation of the system knows itself all the necessary information of a technical nature and gives results with only one command of designer.

In design scenarios the sequence of actions performed by the designer are reflected. The presence of the script allows you to minimize repetitive actions. For example, it is necessary to place and describe a number of similar objects. To do this, the designer once holds the entire sequence of actions, using the entire arsenal of available generic operations, and by identifying this part of the script he may use it again, changing only the input parameters.

The design scenario can have a tree structure where branches are generic operations. At the points of connection of the branches, the control is passed to the designer. There are various options for implementing a design scenario. One project condition scenario can be continued by various generic operations. Or in scenarios one and the same action can be implemented in several generic operations, for example, if there are several calculation methods.

The formation of the project designer works with the CP, which selects from a library of suitable design scenario and launches it for execution. The control processor performs the generic operations scenario, shaping the project. The designer analyzes the generated declarative P data and conducts a dialogue with the CP. In the simplest case, the new project is complete. In general the results of the analysis designer supplements the individual project commands or generic operations, setting up the scenario proposed for a specific project situation. Control processor provides the designer the opportunity to create their own generic operations, create their own design scenarios or edit existing ones. The formation of aggregated transactions is made in the monitor, where from separately manually performed commands generic operations are gathered and the operators in between the K commands, which is the procedural data T, are defined. Thematic coprocessors implement straight and reverse bindings to the Andes formed project (individual or of the generic operations) with the executing processors. Dynamic integration in this information model environment of computer-aided design is implemented in the general case by the control actions of the designer in the process of dialogue with the CP.

Control processor in the information model of the environment of computer-aided design plays a major role. It worked through the design scenario. Script-based control processor decides which commands are given to subject coprocessors to perform generalized operations available in design scenarios.

For the highest quality of failover design scenario the control processor maintains logs of actions performed, whereby the designer at any time has the ability to go back and do all passed actions again.

The Protocol does not duplicate the design scenario. The script provides the designer the ability to perform generic operations in the sequence in which actions will lead to the final result. But in design scenario it is not known how much, for example, were made the same generic operations for such objects. This feature is only available in events protocol.

Each of the thematic coprocessor processes the data of the aspect representation for operations with which it is intended. If he needs the data processed by other thematic coprocessor, he will send an informational message to the control processor. That, in turn, interrupts the first coprocessor and gives the command to the appropriate coprocessor to provide the necessary data in a single information space. After receiving notification of readiness data control processor sends the command to the first coprocessor to continue working. Thematic coprocessors can't apply the same coprocessor directly, for this purpose control processor is used.

Thematic coprocessors can submit only information messages, and receive only commands. Control processor receives only messages and gives commands only. Thematic coprocessors, referring to the control processor, receive from him the command to abort, and continue its work only after receiving the appropriate command from control processor [2]. The use of such model allows controlling actions (Fig. 2).

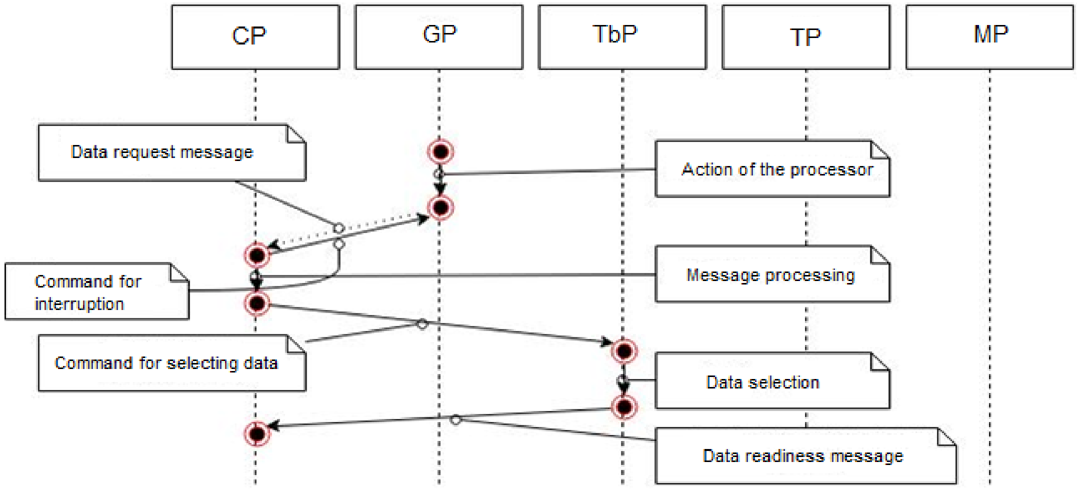

Fig. 2. The scheme of interaction of thematic coprocessors and the CP

Each coprocessor at any point in time provides the designer the possibility to adjust the implementation of the actions of the design scenario. Thematic coprocessors have a number of properties and functions of a service nature. They manage the process of issuing informational messages and respond to commands from the control processor. These functions are particularly important for maintaining the dynamic integration of all processes in the system.

One of the main features of thematic coprocessors is the registration of all actions made by the designer. This makes it possible to create your own generic operations and create certain stages of the design scenario generated by the designer's discretion. To support interaction of all processes occurring in the environment of computer-aided design, theme coprocessors use command processing and parameters transformation. The parameters are converted to an understandable form for every thematic coprocessor included in the environment of CAD, and transmitted in a single information space. The inverse transform of data derived from a single information space allows you to perform actions on objects in accordance with data received from other thematic coprocessors. To support interaction with each other thematic coprocessors have commands transmission of the processed data in a single information space and extract data taken from other coprocessors. Each thematic coprocessor is designed only to perform their strictly defined actions.

Graphics coprocessor performs actions only on graphics [4]. In the process of designing graphics coprocessor occupies an important place, because the drawings are an important part of any design solution. Graphics processor allows you to perform the drawings and to edit existing graphics.

Table coprocessor performs actions only on table data. In the design process it is sometimes difficult to do without table data because the table is a convenient form of hosting large numbers of similar data. Table coprocessor implements the table data by creating and using a database. The designer is constantly accesses the reference data of different nature, which are often presented in table form. One of the results of design are tables of calculated parameters, equipment tables, specifications, etc.

In the process of the project documentation developing text coprocessor is constantly used. It performs actions only on text data. For full performance of the designed object in the project documentation explanatory note is required. To do this, in the design process the designer must constantly create new and amend existing texts.

Text coprocessor has some difference from other thematic coprocessors. Data created in a word processor is mainly descriptive and therefore its impact on other coprocessors is minimal.

In the process of creating the project the math coprocessor can be used, as in project activities mathematical calculations are often required. Mostly the work of the working math coprocessor is invisible. Mathematical formulas and expressions bind numerical values of the commands parameters of various thematic coprocessors. But in situations requiring the choice of a particular method of calculation, math coprocessor provides the designer the ability to control mathematical calculations [2].

The information model allows combining the control actions of the designer, which allows dynamic integration of data in the generalized operations, consisting of commands of various thematic cooperating with one set of declarative information. Such generic operations (created by the designer or available in the library), you can automate dynamic data integration. For the formation of generic operations it is necessary to identify common patterns of connections between procedural data of different (or the same aspects of the presentation.

V. Representation and Interaction of the Data in the Information Model of the Environment CAD

Dynamic data integration of different aspects of the performance is to ensure the flow of data of the same type (integer, real, character, etc.,) between programmatically representing data in different formats. The essence of the method is to link the data of the same type of different aspects of the performance at the right moment of the design process relevant to K operators.

Generic operations that implement dynamic integration of declarative P data are possible only in the presence of operators in between procedural data T. K operators objectively exist in any real project data. Procedural data G, which represent a command of thematic processors, have parameters, representing the project data – declarative P data. The dynamic connection between the thematic commands coprocessors is implemented by K connection operators with binding command parameters [5]. Thus, having regard to procedural data G, it becomes possible to obtain a dynamic integration of declarative data P.

Relations similar to the implemented K operators are used to be called relational. Accordingly, a relational sense has command theme coprocessors, which process procedural data T.

Relational method of dynamic integration is proposed. It is concluded in the use of generic operations, consisting of commands of various thematic coprocessors and integrating data for both one and different aspects of the performance, in the design process [3].

There are general patterns of bindings between procedural data of different or the same aspects of the presentation. Procedural data T are command thematic coprocessors, processing declarative data P of the relevant aspects of the performance.

At the moment of dynamic data integration declarative data R and procedural data T are bound into a single description of the project with the help of K binding operators, i.e., dynamic data integration is the formation of a specific set of K when creating a specific project.

The descriptions of the design object has a state, in which the system can not be automatically solved or has many solutions, which hinders the automation of the design process. To resolve this issue, you may want to impose some restrictions in the K set, which represent the rules of the system formation.

A set of K operators defines the connection between data of one aspect of presentation, or between data of different aspects of the presentation.

Connections are divided into [3]:

– transitive – connections between the data elements are true for both in direct and in reverse sequence;

– non-transitive – connections between the data elements are true in direct and false in the reverse sequence;

– exclusive – connections between data elements are true in direct and non-existent in the reverse sequence.

Consider the possible connection between the data of one aspect of the view [2].

A. The connection between the data of graphic aspect presentation

Data of graphic aspect presentation is used to display graphic information. This aspect of data includes the following properties of a graphic object:

– name of command – declarative data, determining data of a procedural nature;

– how to build – procedural data defining a sequence of execution of the command, which defines how to build the graphical object;

– binding techniques – procedural data, determining how to bind a declarative data of different graphical objects;

– settings – all available declarative data of an object whose value can be changed after construction of the object.

The ways to build graphical objects are different. There are objects with:

– exceptional ways to design when there is only one way of building regardless of the number of parameters of the object (for example, three-dimensional shapes in AutoCAD);

– transitive ways to design that allows you to freely change the sequence of the declarative definition of the object data without changing the object's location (for example, a simple AutoCAD objects: line, point, ray, structural line);

– non-transitive ways to design, when the use of different construction methods without changing declarative data to one object changes its location (for example, complex objects AutoCAD: arc, circle, ellipse).

Methods of objects connection also have certain conditions. For different graphical objects certain ways of binding are applicable:

– exceptional way of binding – building of dependency of one object from another only in a certain binding way (for example, a method of binding the quadrant applies only to an arc, circle, ellipse);

– transitive way of binding – building a dependency of one object from another in the way in which it is possible to change an object dependency to the opposite, applying exactly the same binding way (for example, a binding method for endpoints);

– non-transitive way of binding – building a dependency of one object from another with the method of determining the dependence of the object in only one direction of object hierarchy (for example, the binding of one line to the center of another).

There are situations in which the method of binding directly depends on the construction method. In a certain way of object designing additional data, which after construction are not a part of the object options, can be used. For example, the main parameters of the circle are the center coordinate and radius [3]. A circle can be built by three points that are not represented in the parameters. In such cases it is necessary to use additional data to which object designing is bound.

B. The connection between the data of table aspect presentation

Data of table aspect presentation are used to display information in a table form. Table aspect is represented in the form of a table object with the following properties:

– table name – declarative data which define the path to the contents of a table;

– table row – declarative and procedural data which define the access path to the data contained in the table and the way they are processed;

– table cells – declarative and procedural data which define the access path to the data contained in the table and the way they are processed;

– operations on the data – procedural data defining procedures and commands of table processing.

Operations on the table data are a set of commands database management system performing certain actions when processing tables. Each command, which is part of the operation, performs certain actions on the contents of the table. The command consisting of operations have the ability to bind with other commands, from both its own operations and from the other. There are several ways of binding:

– exceptional way of binding – a command associated with a particular property of the table (for example, used to specify a particular row and table cell);

– transitive way of binding – the property of a single table associated with the property of another, where it is possible to change the dependence of the properties to the opposite by applying exactly the same way of binding (for example, a group of cells in a table associated with the group of cells in another table);

– non-transitive way of binding – building the dependence of one table's properties on another by the method of determining the dependence of the properties in only one direction in the hierarchy of properties (for example, one command from the grid operation is associated with a different operation, for procedural data; and a “one-to-many” connection in a declarative representation of the data).

There are two basic modes of operation with tables: the mode of table modification and the table data reading. In the table editing mode certain data in the cells is recorded, respectively, commands modifying table entries are carried out. In the table read mode only the movement of the pointer across the cells and data reading from the specified cells are carried out [2].

Various modification procedures and reading procedures complicate the work with tables. It is advisable to use different methods for reading and modification in one procedure, by analogy with the properties of objects. The same table procedure has a double effect. In read mode, commands that change the contents of the table are ignored, and a write command is replaced by data reading commands. In read mode and commands modification mode the pointer positioning works the same way.

Application of reading method and modification method impose a condition restricting the freedom of forming table procedures, which is expressed in the transitive reduction and exceptional ways of binding commands within one procedure.

C. *The connection between the data of text aspect presentation*

The data of text aspect presentation is used to display information in text form. Text aspect is represented in the form of a text object with the following properties:

– file name – a declarative data which defines the path of access to the contents of the text file;

– the contents of a text file (text) – declarative data, which contains data necessary to describe the design process and the designed object;

– included fields and links – declarative and procedural data defining the binding of text file declarative data with other software data.

– command – procedural data which defines the procedures and word processors commands for creating and formatting the text.

Included fields and links required for binding the declarative data of a text aspect presentation with declarative data as both text and other aspects of the performance. Each included field or link to an object changes the meaning of the text. Using fields and links the connection between declarative data from various software processing data relevant to aspects of representations is implemented. There are several ways of binding:

– exceptional way of binding – in the text link to some object is inserted whose data is used in the text and the content change of a text file has no effect on the contents of the binding object (for example, insert picture, formulas or labels);

– transitive way of binding – link binding with external object, and modification of data is possible both in the text and embedded object (for example, connection by inserting object);

– non-transitive way of binding – binding of text information with this or any other text file parameters when the associated text is completely dependent on the inserted parameter (for example, inserting the value of the current date, time, settings, document, etc).

Commands of word processor are designed to perform operations on the document, the text, included fields and links, and for communicating of word processor with other software as well. The command processor can be combined into sets to perform certain operations, for brevity let's call the sets of commands – operations. In some situations, you can run the commands separately [2].

The binding of commands of word processor can be performed between operations, between the individual commands that are not included in the transaction and between the individual commands and operations. You can use the connection between the commands included in the transactions with certain commands, operations or commands included in the other operations.

The connection between the operations is performed in the following ways:

– exceptional way of binding – the operation is associated with another specific operation, and these operations are fully dependent on each other (for example, the operation of opening the document macros associated solely with the operation of opening a text document);

– transitive way of binding – the connection between operations when the first of the related operations affect the second, and vice versa execution of the second affects the first (for example, output the same text data in different details of the document or in different formats);

– non-transitive way of binding – building according to one operation from another, a way of determining the dependence of the operations in only one direction in the hierarchy (for example, the operation of filling of the cells embedded in the text of the table depends on the data receive operation).

The connection between the individual commands that are not included in the operation is performed in the following ways [4]:

– exceptional way of binding – the command binds with the other specific command, these commands can only communicate among themselves and not with any other (for example, copy the text in the area of data exchange and command inserts data);

– transitive way of binding – the connection between the commands, when related commands can be executed in any order, affecting the associated command (for example, the formatting interconnected commands on the model, can be performed in any sequence, formatting paragraphs by one);

– non-transitive way of binding – building the dependence of one command from the other one by determining the dependence of the commands in only one direction of the hierarchy.

The connection between the individual commands that are not included in the operations and integer operations are performed in the following ways:

– exceptional way of binding – a command associated with a transaction, and the connection can be performed only between them and not with any other commands or operations (e.g., the search operation is used to open the document macros can only be associated with the command opening the document);

– transitive way of binding – the bond between a command and the operation when the related commands and operations can be performed in any order, affecting each other (for example, the commands formatting paragraphs for example);

– non-transitive way of biding – building the dependence of operation from the command by the method of determining the dependence of the operations from command in only one direction of the hierarchy.

Commands included in the operation have the ability to communicate both with separate commands and with integer operations. When binding commands included in operations the same methods of connection, as for individual commands are applied. Because these commands already have bonds with commands of the same operation, methods of binding these commands will have some restrictions.

Command included in the operation, may contact subordinating bond only if there are available from binding of command parameters. Subordinating connection can only be carried out by exceptional or non-transitive means of connection. When performed the operation, which has in its composition the associated command will be executed only after the full execution of other related operations. If the command, included in the operation, associates with the command, which is part of another operation with a cyclic repetition of commands, the operation, including the associated command will be executed as many times as other related operation.

D. The connection between the data of mathematical aspect of representation

The data of mathematical aspect of representation is used to display mathematical information. Using mathematical expressions and formulas, the required in the design process values calculations are carried out. This aspect of view includes the following objects of the mathematical properties of the processor [3]:

– input parameters – declarative and procedural data representing the argument of the function or formula;

– output values – declarative and procedural data which is the result of a function or formula.

The work of mathematical calculations is a substitution for required values as arguments of mathematical function, and the results of calculations. For various kinds of calculations a formula and or a set of interrelated expressions can be used. The binding of expressions is performed by using declarative components of arguments and output values. Each parameter of a mathematical expression is written in a certain character set, which is used as argument or output values depending on the purpose of the formula.

Ways of mathematical expressions binding have some feature that distinguishes them from the previous methods of binding data from different aspects of the performance. Output values can be used only for lookup as arguments of mathematical expressions. In this situation there is no transitive way of binding, as the first function can not be influence by the calculations produced by the second function. Thus, there are two methods of mathematical objects binding:

– exceptional way of binding – the mathematical expression for its output value is associated with an argument of other mathematical expressions and character expression that specifies the output value of the mathematical expression is used nowhere (for example, the use of intermediate functions for a series of calculations);

– non-transitive way of binding – the argument is one of the mathematical expressions associated with the output of another, and the argument can use the output value of only one mathematical expression, and one output value can be related to the arguments of several mathematical expressions.

E. The connection between different data aspects of the presentation

The relation between these different aspects of the presentation is determined by a set of K connection operators. The types of data relation of various aspects of the performance between them are expressed in the form of binding of procedural components of the data. Due to the difference of data binding methods from different aspects of the performance for each pair of aspects personal operators from the set K are applied [2].

For data of the graphical aspect presentation it is possible to establish a subordinate relation to other aspects of data representation. In connection with table data, there are three ways of binding:

– transitive way of binding – binding one parameter of the graphics commands directly from the command select columns or table rows from the table operation;

– non-transitive way of binding – binding graphic option commands with selecting values commands from a table;

– exceptional way of binding – binding commands GPU-valued entire operation, allowing to execute the graphics command only after performing table operations.

When binding graphics commands with a mathematical processor there are only two possible ways of data binding:

– non-transitive way of binding – binding one parameter of the graphic command with sets of commands and matrix math package processing;

– exceptional way of binding – binding one parameter of the graphic command with a specific output value of the mathematical expression.

For binding graphics commands with the word command processor three methods of binding is provided:

– transitive way of binding – graphical command associated with the operation or word processor command and the binding method is the possibility of mutual influence of connected command parameters with each other;

– non-transitive way of binding – graphical command associated with the operation or word processor command and the performance of the graphics commands is only possible after the execution of the operations or word command processor;

– exceptional way of binding – the graphics command is associated only with a particular transaction or word command processor and the execution of related commands and operations can be in any order [4].

For data of the table aspect presentation it is possible to establish a subordinate relation to other aspects of data presentation. When binding with the graphics data three way binding are used:

– transitive way of binding – binding commands select columns or table rows from the table transactions directly with one parameter of the graphics command;

– non-transitive way of binding – binding commands insert values into a table with the graphic option commands;

– exceptional way of binding – binding operations of table processor which allow to perform table only operation after you run graphics operations.

For binding mathematical data only two ways of binding are used:

– non-transitive way of binding — binding commands or insert values into a table with the command processing matrices or sets of mathematical package;

– exceptional way of binding – binding commands or insert values into a table with a specific initial value of the mathematical expression.

For binding text data, there are three means of connection:

– transitive way of binding – table command or operation associated with the operation or word command processor and in this connection method mutual influence of associated command parameters at each other is possible;

– non-transitive way of binding – table command or operation associated with the operation or command word processor, and the performance of the graphics commands is only possible after the execution of the operations or command word processor;

– exceptional way of binding – table command or operation is associated only with a particular transaction or word command processor and the execution of related commands and operations can be in any order.

For mathematical aspects of data presentation downstream bindings with other aspects of the data can be installed [2]. Due to the nature of the mathematical aspects of data presentation for binding with other aspects, you cannot use a transitive way of binding. When communicating with the graphics data, there are two ways of binding:

– non-transitive way of binding – binding process of commands sets with a single parameter of the graphics command;

– exceptional way of binding – binding argument of a mathematical expression with a specific parameter of the graphics command.

For connection with table data are used in two ways:

– non-transitive way of binding – binding process of command matrices or sets of commands with command of rows and table cells selecting;

– exceptional way of binding – binding argument of a mathematical expression with the command of values from a table selecting.

The feature of the word processor as a display tool of project data imposes on the binding operator K between data of mathematical and text aspects the restriction on the use of different methods of connection [3]. For the data connection of mathematical aspect presentation to the text aspect of subordinating connection it is not possible to use a transitive way of binding. Non-transitive way of binding acquires properties of exceptional way of binding. Therefore, for binding with the data of text aspect presentation exceptional way binding is used – binding of argument of a mathematical expression with a particular command or operation of the word processor.

For the text aspect of the present data it is possible to establish a subordinate relation to other aspects of data representation. For binding commands and operations of word processor with graphics commands three means of binding are represented:

– transitive way of binding – operation or word command processor binds with the graphics command, and in this way of binding the possibility of mutual influence of associated command parameters at each other is performed;

– non-transitive way of binding — operation or word command processor only binds with the graphics command, and execution of related operation or word command processor is only possible after the execution of the graphics commands;

– exceptional way of binding – text operation or command connected only with the particular graphics command, and the execution of related commands and operations can be done in any order.

In connection with table data, there are three ways of binding:

– transitive way of binding – the connection of one operation parameter or a command of word processor directly with commands selecting columns or table rows from table operations;

– non-transitive way of binding – the binding of operation parameter or a command of word processor with commands of selection from a table;

– exceptional way of binding – the binding of action, or a command of word processor with an operation table in whole, allowing to perform an operation or word command processor only after performing table operations.

When connecting operations or word command processor with mathematical processor only two ways of data binding are possible:

– non-transitive way of binding – binding of one operation parameter or word command processor with processing commands sets and matrix math package;

– exceptional way of binding – binding one operation parameter or word command processor with a specific output value of the mathematical expression.

Many operators R consist of the subsets relevant to a particular aspect of the data representation.

Thus, each subset of relations of a particular aspect of the presentation data is divided into groups of relations "inside" and "all other".

On the types of connections thematic subsets of the operators are grouped as follows:

– transitive;

– non-transitive;

– exceptional.

Sets contain different stack of subsets, which is due to a fundamental lack of certain kinds of connections between these different aspects of the performance [5].

The above formal restrictions allow to automate the process control generated by the designer of the relations between the data. This will reduce the cost of dynamic data integration, eliminating formal design errors.

In the real set R there are semantic restrictions on the dynamic integration of data, which identifies the designer. For example, restrictions of this type: "formally, nothing prevents to portray the motor in a particular place in the framework of the power-supply system, but that this scheme is not necessary for it to work". In a sense, the work of the designer is to select the desired option.

CONCLUSIONS

It is proved the necessity of integrated computer-aided design systems. The information model of computer-aided design environment is considered. It is researched the different aspects of this problem.

REFERENCES

[1] Lee, K. CAD Basics (CAD/CMA/CAE), Peter Press, 2004.

[2] Kupriyanov, V. V.; Pechenkin, O. Yu.; Suslov, M. L. CAD and artificial intelligence systems, ROSNY and IT UP: DBMS, 1995. (in Russian).

[3] Berezhnoj, G. Problems building large IT systems, PCworld, 1998. (in Russian).

[4] Norenkov, I. P. Basics of computer-aided design. Peter Press, 2002. (in Russian).

[5] Kristi Morton. Dynamic Workload Driven Data Integration U. of Washington, 2012.

Received 02 September 2014.

Sineglazov Viktor. Doctor of Engineering. Professor.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant.
Publications: more than 450 papers.
E-mail: svm@nau.edu.ua

Godny Andrew. Ph. D. Student.
Education: National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine (2014).
Research interests: systems and process control, system identification, automatic control of industrial processes.
Publications: 2.

E-mail: andrewgodny@gmail.com

В. М. Синєглазов, А. П. Годний. Інформаційна модель середовища автоматизованого проектування
Розглянуто інформаційну модель середовища автоматизованого проектування. Інформаційна модель забезпечує можливість створення сценарію проектування.
Ключові слова: динамічна інтеграція; система автоматизованого проектування; система управління базами даних.

Синєглазов Віктор Михайлович. Доктор технічних наук. Професор.
Кафедра авіаційних комп’ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки.
Кількість публікацій: більше 450 наукових робіт.
E-mail: svm@nau.edu.ua

Годний Андрій Павлович. Студент.
Освіта: Національний технічний університет України «Київський політехнічний інститут», Україна (2014).
Напрямок наукової діяльності: ідентифікація систем управління, цифрові системи управління
Кількість публікації: 2.
E-mail: andrewgodny@gmail.com

В. М. Синеглазов, А. П. Годний. Информационная модель среды автоматизированного проектирования
Рассмотрена информационная модель среды автоматизированного проектирования. Информационная модель обеспечивает возможность создания сценария проектирования.
Ключевые слова: динамическая интеграция; система автоматизированного проектирования; система управления базами данных.

Синеглазов Виктор Михайлович. Доктор технических наук. Профессор.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Киевский политехнический институт, Киев, Украина (1973).
Направление научной деятельности: аэронавигация, управление воздушным движением, идентификация сложных систем, ветроэнергетические установки.
Количество публикаций: более 450 научных работ.
E-mail: svm@nau.edu.ua

Годный Андрей Павлович. Студент.
Образование: Национальный технический университет Украины «Киевский политехнический институт» (2014).
Направление научной деятельности: системы и процессы управления, идентификация, автоматизация систем управления технологическими процессами.
Количество публикаций: 2.
E-mail: andrewgodny@gmail.com