UDC 681.5.015 (045)

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

MEANS OF DYNAMIC INTEGRATION DATA REALIZATION AT THE LEVEL OF SEPARATE COMPONENTS OF THE CAD ENVIRONMENT

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

Abstract**—***The structure of computer-aided design system based on dynamic data integration consisted of a control processor and thematic coprocessors: graphic coprocessor, table coprocessor, math coprocessor and text coprocessor is substantiated. The construction of separate elements is considered.*

Index Terms—Data dynamic integration; computer-aided design system.

I. INTRODUCTION

The method of data dynamic integration is concluded in the connection of one command or group of commands of one software tool, the executing result of which is the meaning of a specific data type, with commands of other software tool in a unified information process. Both considered software tools have equal rights [1]. The order of interaction is determined by conditions of information process and user requirements. At the data dynamic integration the system operates the commands forming objects parameters, and implements objects connection directly between the commands, while providing a more flexible way of combining and "understanding" of different types data. The parameter of any object in the system should not be separated from the command. This parameter is only a formal representation of data in the system.

II. STRUCTURAL SCHEME DEVELOPMENT OF INTEGRATED COMPUTER-AIDED DESIGN SYSTEM

Computer-Aided Design (CAD) system using the method of data dynamic integration is a structure consisting of a control processor (CP) and thematic coprocessors: graphic coprocessor (GP), table coprocessor (TbP), math coprocessor (MP) and text coprocessor (TP). The design process is ensured by the design scenario by the control processor. The designing scenario is a set of generic operations that can be represented as a graph [2]. The generic operation consists of multiple commands of thematic coprocessors with a common semantic completeness.

The presented software implementation model of the CAD environment most closely matches the problems solution, entrusted to the system for data dynamic integration providing. In the present implementation it is realized the complete processing of all generic operations available in the designing scenario. A structure that fulfills a "command-message" coupling in all thematic coprocessors has been created. The mechanism of data transmission between the processors has been implemented (Fig. 1).

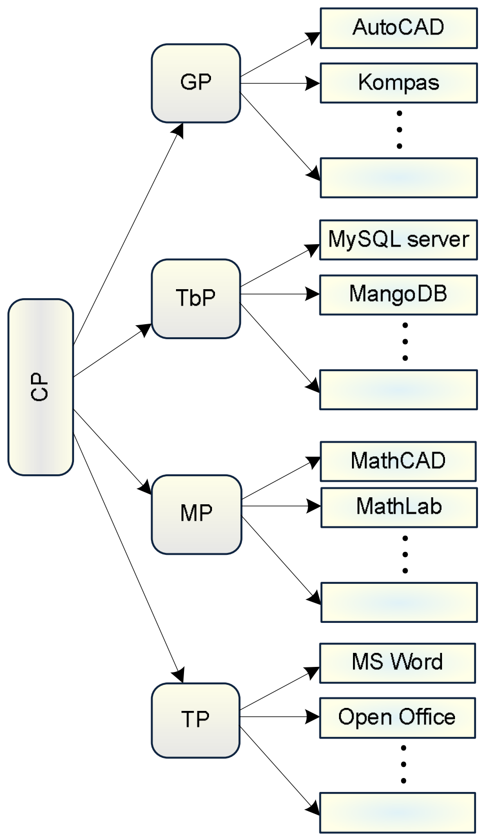

Fig. 1. Block diagram of computer-aided design environment

III. DESCRIPTION OF COMPUTER-AIDED DESIGN ENVIRONMENT

Control processor

Control processor supports execution of generic operations according to the design scenario [1]. Control processor issues commands to thematic coprocessors and accepts from them only informational messages. Each generic operation action starts with the CP command. Through the means of communication command is transmitted to the thematic coprocessors, where it is processed accordingly. After completing all steps of the command thematic coprocessors return control to the CP and send it the information message through the means of communication.

The control processor consists of the following modules: control module, design scenario module, events logging module, message processing module, command processing module, unified information space support module, communication module (Fig. 2).

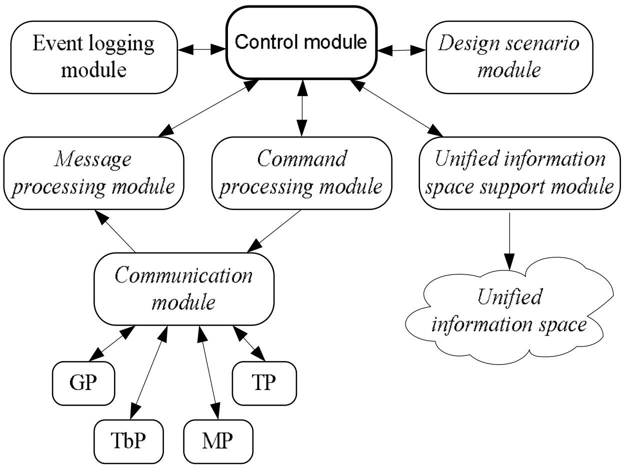

Fig. 2. Structural diagram of the control processor

Control module

The control module regulates the relationship between modules inside control processor. The module performs and supports the following functions: redistribution of data from one module to another, surveying the state of the modules, determining task priority, transferring control to the modules in accordance with a certain priority.

Design scenario module

Design scenario module is responsible for handling the design of the script file [3]. Module performs and supports the following functions: viewing and editing the file as text, viewing and editing the file in the form of graph, reading commands from a file and transferring them to the control module, receiving messages from the control module.

It outputs to the control module the following command on the script design.

It receives from the control module the message of the completion of the previous command.

Event logging module

The event logging module is responsible for recording all the actions made in the event log file. The module performs and supports the following functions: viewing the event log file, processing the event log file.

It outputs to the control module message with results of command execution.

It receives from the control module command on including some events in the log file.

Message processing module

The message processing module is responsible for receiving messages from the thematic coprocessors through the communication module. The module performs and supports the following functions: receiving text messages from the communication module, accumulation of messages in the buffer in the order of their arrival, issuing messages to the control module on request.

It outputs to the control module the next received message from the thematic coprocessors.

It receives from the control module a request for the transfer of messages from the coprocessor, and from module of communication it receives text messages from thematic coprocessors.

Command processing module

The command processing module is responsible for transmitting commands for thematic coprocessors to communication module. The module performs and supports following functions: processing commands for thematic coprocessor module from design scenario module, partitioning of generic operations into commands and transferring them to their destinations (thematic coprocessors), transmission of text commands to the communication module.

It outputs to the communication module text commands for thematic coprocessor modules, to the communication unit it outputs the text message sending command to the communication module.

It receives from the control module the text of commands from a script design.

Unified information space support module

The module provides unified information space support and controls access to it of thematic coprocessors [5]. The module performs and supports the following features: containing the computer's memory and physical space for entering data into it, regulating access to the space for theme coprocessor, providing data storage to their demand, removing unnecessary data from the unified information space (UIS), reading and writing overhead in the UIS.

It outputs to the control module the overhead from the UIS and overhead to UIS to be recorded.

It receives from the control module overhead to write to the UIS and overhead from UIS to be sent to the control module.

Communication module

The communication module is responsible for liaising with the thematic coprocessors. The module performs and supports the following functions [3]: ensuring communication with the thematic coprocessors, receiving text of commands to be sent to the thematic coprocessors, sending commands to the thematic coprocessors, regulation of coprocessors access to the control processor, receiving messages from the thematic coprocessors and transferring them to the message processing module.

It outputs to the thematic coprocessors text of commands and to the message processing module it outputs the text of the news reports.

It receives from the command processing module text of commands for thematic coprocessors, and from coprocessors it receives text of information messages.

Graphics coprocessor

The graphics coprocessor is a graphics processing tool. GP provides: communication with graphic editors via the driver D, performing commands from CP, information transmission to the CP, issuing commands for graphics editor, receiving informational reports from graphic editor.

The graphic coprocessor implements the execution of graphic works of the design process and creates connection between CP and graphics execution modules.

Tabular coprocessor

Tabular coprocessor represents tabular data processing means [2]. Table coprocessor provides: connection to the database through the driver D, performing commands from CP, information transmission to the CP, issuing commands to the DBMS, receiving informational messages from DBMS.

The table coprocessor implements the interaction of the database with the design process. It supports management of internal business databases.

Math coprocessor

The math coprocessor means to perform mathematical operations. MP provides: communication with the mathematical editor through the driver D, CP command execution, information transmission to the CP, issuing commands to the mathematical editor, receiving informational messages from math editor.

Math coprocessor implements the execution of mathematical operations arising in the design and supports the interaction of mathematical performing processors with CP.

Text coprocessor

The text coprocessor is a text data processing module [2]. Text coprocessor provides: communication with a text editor through the driver D, executes commands from CP, information transmission to the CP, issuing commands to the text editor, receiving informational messages from word processor.

The text coprocessor implements a text description of the designed facilities, used equipment, performed calculations, etc. The text coprocessor supports interaction between word processors and CP.

Therefore, the structure described above of the server node of integrated computer-aided design software implementation demonstrates the ability of the dynamic relational data integration at the level of the control processor. Presented server node has the ability to perform the basic functions of the proposed control processor: planning scenario management, command transmission and reception of messages from the theme coprocessors, logging designer activities.

Thematic coprocessor

The thematic coprocessor represents a tool to process data of their own representation format. The thematic coprocessor does not operate directly on the data in the files of the project and on the screen. The thematic coprocessor only transmits certain commands accordingly to the Executable processors (EP). The executable processors are connected to the thematic coprocessor through the appropriate drivers. The thematic coprocessor performs data translation from the general system language to the language of the Executable processor. Through the communication facilities thematic coprocessor sends commands to EP and receives only informational messages.

The coprocessor provides reports of the implementation of commands for the control processor, and for the executable processor it provides the commands to execute.

The coprocessor receives from the control processor instructions for performing by EP, and from executable processors it receives information messages.

The thematic coprocessors consist of the following modules: control module, module of communication with CP, module of outgoing messages processing, module of incoming commands processing, module of connection with the EP, module of outbound command processing, module of incoming messages processing, module of unified information space support, drivers.

Control module

The control module regulates the relationship between the thematic coprocessors [4]. The module performs and supports following functions: redistribution of data from one module to another, surveying the state of the modules, the definition of task's priority, and transferring control to the modules in accordance with a certain priority.

It outputs to the outgoing messages processing module a request for the issuance of a command from the CP module, to the processor of incoming commands it outputs a request for the issuance of a command from the control processor, to the processing module of outgoing commands it outputs the creation of a the commands for CP, to the module for processing incoming messages it outputs a request for the transfer of information, to the unified information space module it outputs data for writing into a unified information space, and to the driver it outputs proprietary information about the nature of the work executed by the processor.

It receives from the outgoing messages processing module message about the implementation of the command, from the incoming commands processing module it receives a command from the control processor, from the outgoing commands processing module it receives information about the implementation of the command, from the incoming messages processing module it receives processed information message, from the unified information space support module it receives data from a unified information space, and from drivers it receives proprietary information about the nature of the work executed by the processor.

Communication module

The communication module with CP is responsible for liaising with the control processor [2]. The module performs and supports the following functions: providing connection with CP, receiving text information messages sent to the CP, sending an informational message to the CP, receiving commands from CP and transferring them to the module for incoming commands processing, accumulation of information messages in the buffer as they arrive.

It outputs to the control processor reports on the implementation of the commands by thematic coprocessor, and to the module for processing incoming commands it outputs commands received from CP.

It receives from the control processor instructions to perform for the thematic coprocessor module, and from the module of outbound messages it receives text information reports about command implementation by the thematic coprocessor.

Executable processors communication module

The executable processors communication module enables communication with executable processor. The module supports and performs the following functions: ensuring connection with EP, receiving text messages from EP, the transferring of text messages in the information incoming messages processing module, receiving instructions from the outgoing command processing module and transferring them to the EP.

Unified information space support module

The module provides access to a unified information space for thematic coprocessors. The module performs and supports following functions: regulating access to a unified information space, providing the reception and transmission of data to the thematic coprocessor from a unified information space, reading and writing data to UIS.

Executable processor

The executable processor is a software product that provides the performance of certain actions on the object design [4]. The executable processors are divided into several groups according to the nature of interaction with the thematic coprocessors. Each executable processor should meet the following conditions: ability to communicate with other software, having means of internal programming.

The executable processor connects to the thematic coprocessor via appropriate driver, which provides a combination of commands that can be executed by the processor and thematic coprocessor.

The executable processor consists of the following modules [1]: control module, communication module, unified information space support module, means of execution.

Unified information space

The unified information space represents structured records of pointer type. The processors place data at the top of the queue, where they are accumulated, and appropriate processors take data from the bottom of the queue on demand. Thus, using the queue mechanism the problem of obtaining the data needed for processors to work with can be solved.

To solve the problem of storage of any data type in a unified information space, we use pointer type processing mechanism. The pointer type variable contains only the address of the memory that stores the needed record [5]. By converting pointer data to the requested type the processors have access to the type of data that they need at the moment. The design scenario is responsible for data completeness.

Since in the process of data exchanging between the processors it is necessary to pass multiple values at the same time it is advisable to use structured record [3]. To pass multiple data via a single variable in a unified information space, you must perform two commands of the CP and get reports from CP. Thus, by using one variable four processor cycles are utilized. When transmitting n values, the number of processor cycles required to transfer data increases respectively to the $n - 1$ times. Therefore, it is advisable to use a structured record of several variables, where the number of variables is greater than or equal to n.

CONCLUSION

Above is presented the structure of thematic coprocessors and executable processors of the integrated CAD system implementation that demonstrates the ability of the dynamic integration of relational data. Presented software components are able to perform the basic functions of the proposed thematic coprocessors: performing generic operations of design scenarios, receiving commands and sending messages from the control processor and logging activities.

REFERENCES

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

[2] Pozin, B. A. (1995). Modern software engineering tools for the creation of open information systems applications, ROSNY and IT UP: DBMS.

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

[4] Soulless, A. N.; Zhizhchenko, A. B.; Kulagin M. V. and Serebryakov, V. A. (2000). Integrated. Resources Information System of Sciences and Technology Digital Library.

[5] AutoCAD DWG http://en.wikipedia.org/wiki/DWG

Received 20 April 2015

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 500 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: 3.
E-mail: andrewgodny@gmail.com

В. М. Синєглазов, А. П. Годний. Засоби реалізації динамічної інтеграції даних на рівні окремих компонентів середовища автоматизованого проектування

Обгрунтовано структуру автоматизованої системи проектування, яку розроблено з використанням методу динамічної інтеграції даних, що складається з керуючого процесора і тематичних сопроцесорів (графічного, табличного, математичного та текстового). Розглянуто побудову окремих її елементів.

Ключові слова: динамічна інтеграція даних; автоматизована система проектування.

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 500 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 публікацій: 3
E-mail: andrewgodny@gmail.com

В. М. Синеглазов, А. П. Годний. Средства реализации динамической интеграции данных на уровне отдельных компонентов среды автоматизированного проектирования

Обоснована структура автоматизированной системы проектирования, разработанная с использованием метода динамической интеграции данных, которая состоит из управляющего процессора и тематических сопроцессоров (графического, табличного, математического и текстового). Рассмотрено построение отдельных ее элементов.

Ключевые слова: динамическая интеграция данных; автоматизированная система проектирования.

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 500 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: 3.

E-mail: andrewgodny@gmail.com