UDC 681.5.015 (045)

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

DYNAMIC DATA INTEGRATION IN THE DESIGN OF COMPLEX CAD SYSTEMS

Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua, [2]andrewgodny@gmail.com

***Abstract*—*Presented technology of multimethod environment for optimal design. In order to ensure flexibility and adaptability of the algorithmic framework is proposed to use a modular component-based approach to structuring optimization problems and algorithms for solving them. This algorithmic software in general is defined as a set of libraries and optimal selection procedures intelligent interconnecting modules depending on the specific task at hand.*

Index Terms—Dynamic integration; Computer-Aided Design; integrated environment; design; optimal solutions; algorithmic framework.

I. Introduction

The problems associated with the adoption of optimal solutions, occupy an important place in computer-aided design. Multidimensional nature and variety of optimization problems in Computer-Aided Design (CAD), their close relationship with the tasks of modeling and analysis required to improve the methods and tools to support the processes of optimal design at different stages.

In accordance with Power Standards, CAD structure consists of the following elements (Fig. 1):

- Hardware of CAD (HCAD) is a set of tools for CAD;
- CAD subsystems, as part of the CAD structure, occur when users operate HCAD of CAD subsystems;
- HCAD subsystems of CAD are set of software & technical complexes (STC), software methodical complexes (SMC) and selected CAD software components that are not included in the software systems, united by a common subsystem function;
- software components of STC;
- software components of SMC;
- software components of CAD, not included in SMC and STC.

The set of the various HCAD of subsystems form HCAD of entire CAD in general.

Organization for optimal solutions search procedures as part of STC in modern CAD systems is complicated by the steady increase in both the number of supported optimization procedures, as well as the increasing complexity of the structure of CAD themselves [1]. Typical structure of STC is shown in Fig. 2. This structure increases the time and cost of designing and building such systems. This problem is particularly evident in the design of complex systems, the functioning of all the scenarios that cannot be taken into account, which leads to an increase in the number of bugs in the software. The complexity of evaluating the correctness of these systems limits the ability to use standard approaches to building optimization algorithms that are used in traditional CAD, which ultimately reduces the efficiency of the process of optimum design. In addition, features include a variety of productions of complex systems design problems, high dimensionality, the multiplicity of technical and economic requirements of the basic characteristics, significant labor input of simulation and analysis, consider all of that as part of the standard software of CAD becomes impossible.

II. Problem Statement

This scheme of STC in the development of integrated CAD system is transformed into a set of algorithms, resulting in a lack of flexibility and adaptability of the system. This approach greatly complicates the changes and implementation of new algorithms into the system.

It is necessary to develop a method of designing algorithmic modules of integrated CAD, allowing to organize flexible complex optimization algorithms with the possibility of integration into a single information space of complex CAD. The objectives to be achieved:

- decomposition of algorithms in STC;
- allocation of the common elements of the algorithms for the modules in a separate component, which will be available to all STC modules;
- merge decomposed algorithms of the module in a single core, which will increase the flexibility of the modules and will open the possibility to configure the algorithms.

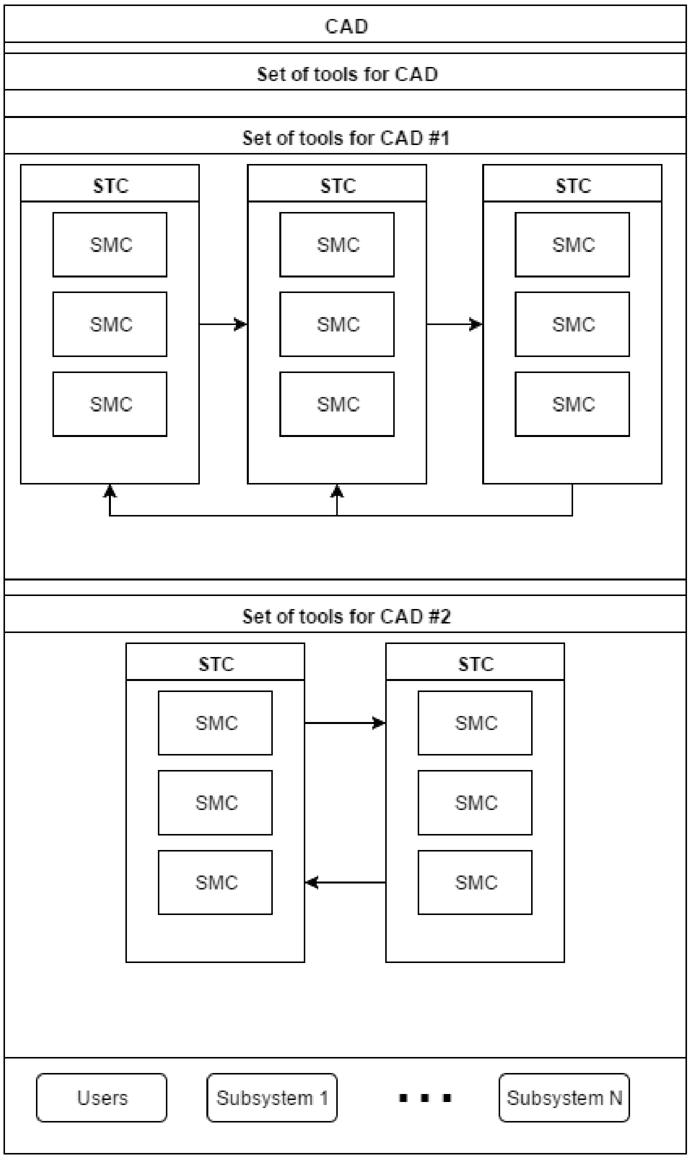

Fig. 1. Computer-Aided Design structure

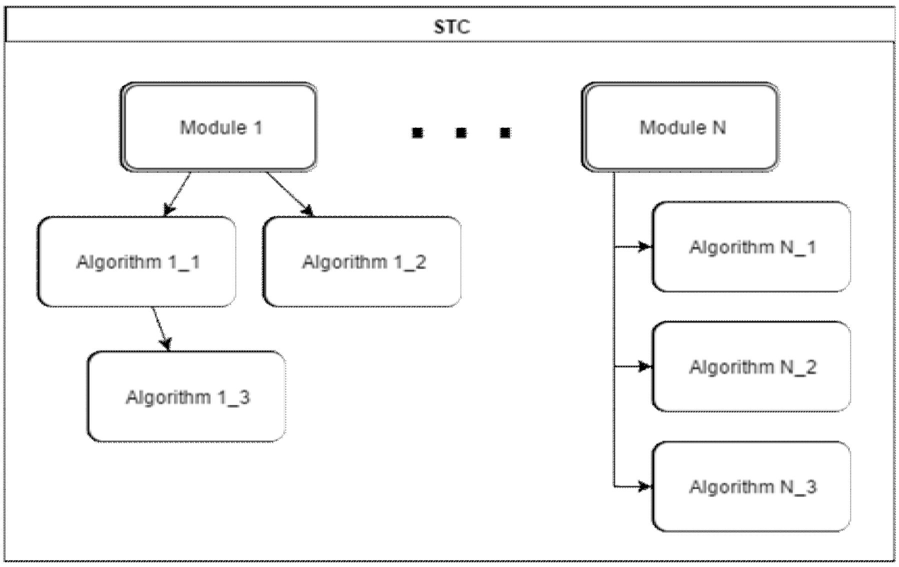

Fig. 2. Software & technical complexes structure

III. The Method of Constructing Algorithmic Models

To date, two approaches are used for the creation of algorithmic software for complex integrated systems:

- Component approach;
- Modular approach.

Component approach is essentially based on the concept of component – an independent module, intended for re-use and deploy, is realized as a set of language constructs, united by a common feature, and organized in accordance with certain rules and restrictions [3].

The modular approach is the organization of the algorithm as a set of small independent units, called modules, the structure and behavior of which are subject to certain rules [3]. Using a modular approach simplifies the testing of the library and error detection. Hardware-dependent subtasks can be strictly separated from other subtasks that improves mobility of created programs. The role of the modules can be played as a separate component of the algorithm, as well as a group of individual algorithms that implement some of the functionality and provides an interface to it.

The solution of the problem above can be achieved by using component-integrated modular approach to the design of complex objects. This approach involves the construction of a complex optimization algorithms to ensure alignment process more complete formalization of the problem with its decision, and their union, together with the procedures of multiple modeling in intelligent adaptive multimethod environment with the ability to dynamically adjust to the various classes of problems of optimum design.

IV. Structure Design of Optimal CAD Software Module

When designing complex systems with a wide range of optimal design algorithms the important role is played by the use of dynamic data integration. The method of dynamic data integration based on a combination of modules that perform the same type of problem in one subsystem for the purpose of adaptive change in the design, based on dynamic assessment of the properties of tasks [5]. To solve the problem is offered to represent the algorithmic base of complex systems as a set of modules to form an optimal selection and adaptive procedures to ensure the formation of numerical schemes applied to different situations.

The implementation of component-modular approach via the organization of the algorithmic base of optimal design involves structuring optimization problems and methods for solving them. Included in the software, algorithmic optimization procedures are divided into a number of functional components, realized by one or more modules, various combinations of which result in the formation of specific optimization procedures. This algorithmic basis of optimal design of complex systems can be defined as a set of modules depending on the specific task at hand.

The module is constructed in a hierarchical manner, the solution provides optimal design problems of varying complexity, and includes procedures for the two classes: scalar unconstrained optimization procedures; problem-oriented plug-ins. Unconstrained optimization procedures form the core of the algorithmic base [4]. Connection of external modules is carried out at the external level, depending on the specific problem to be solved (the space dimension of variable parameters, the character changes in the parameters, the number and types of direct and functional limitations, the number of optimality criteria, etc.) and does not require changes to the algorithmic core.

Each module is stored separately and represents independently compiled unit and has two names: the external, announced the title of the module and coincides with the name of the file containing it, and the interior, which is heading of the implemented procedure. Alternative interchangeable modules have different external name and identical internal names. At the same time internal name is associated not with a specific object (module), but with its function that allows you to use in different situations modules, similar in purpose but with different implementation strategies.

The structure of the algorithmic modules library generally can be represented in the form of aggregated graph, whose vertices correspond to the simple graphs with specific functional purpose. Thus various algorithmic structures can be obtained by selecting a specific path in the graph. Then, a further specification of the computational scheme is done by disclosing aggregated graph vertices included in the generic algorithm.

V. Building a Kernel of Algorithmic Module

Algorithms of optimal design is structured as follows: procedure of continuous, discrete and discrete-continuous optimization [2]. At the same time continuous and discrete-continuous optimization procedures designed to solve the problems of parametric synthesis, and discrete optimization procedures – how to solve the problems of parametric and structural synthesis.

Continuous search engine optimization modules can be divided into two groups: modules that implement adapted randomized algorithms of local optimization; module-wise search. Random search algorithms and their adaptive extensions formed on the basis of a combination of modules of the first and second groups [4]. Other modules are optional and can be connected to the search procedures as needed. The modular structure of the continuous optimization algorithms can be collectively represented as a directed graph, whose vertices correspond to algorithmic modules and arcs – possible links between them (example on Fig. 3). Specific computational scheme formed by "assembling" modules in one algorithm (or by selecting a particular path in the graph). Through various combinations of modules we can generate both standard and original search optimization algorithms. Thus, based on a limited set of invariant modules we can build a different degree of difficulty and content optimization procedures.

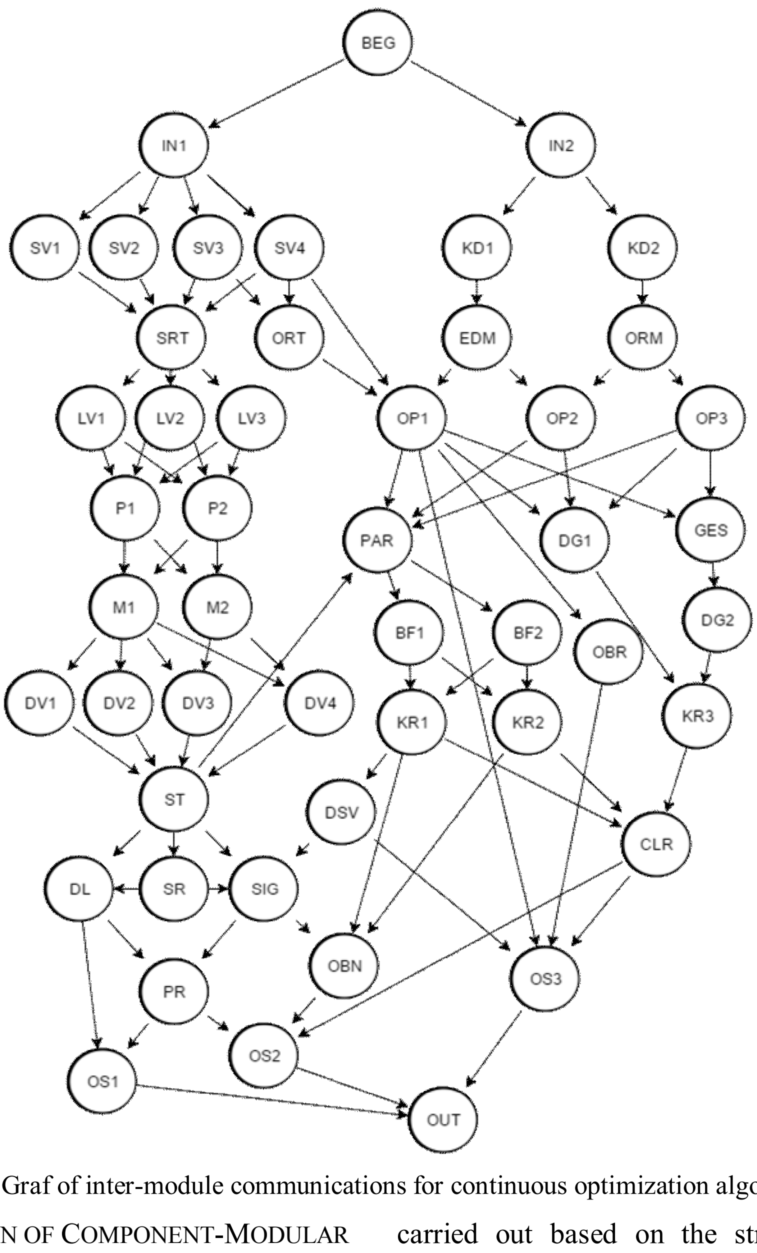

Fig. 3. Graf of inter-module communications for continuous optimization algorithms

VI. IMPLEMENTATION OF COMPONENT-MODULAR APPROACH IN DESIGNING OF ALGORITHMIC MODULE

Consider the example of construction of adaptive algorithms library for uninterrupted optimization using component-modular approach. Formation of algorithmic modules for continuous optimization carried out based on the structuring of adaptive search procedures [3]. On Fig. 3 optimization procedures are presented in the context of component-modular approach, using dynamic data integration. Algorithms are divided into modules. The function of algorithmic models is briefly reviewed in Table I.

TABLE I

THE FUNCTION OF ALGORITHMIC MODELS

Modules	Designation
BEG	The initial approximation is entered by the user in an interactive mode.
IN1 IN2	Installation procedures of algorithmic parameters
Adaptive procedure for local optimization	
SV1 SV2 SV3 SV4	Production of random vectors
ORT	The orthogonalization of the vectors
SRT	Arranging of implementations
LV1 LV2 LV3	Determining the level of the objective function
P1 P2	The calculation of probabilities to determine the mathematical expectation
M1 M2	The calculation of the mathematical expectation of random vectors
DV1 DV2 DV3 DV4	Stir in random vectors
ST	Setting the search step
OS1 OS2 OS3	Checking convergence of algorithmic procedures
OBN	Update samples
SR	Homogenization of random movements
PR	Forecasting the global optimum
Wise search procedures	
KD1 KD2	Select search trends in coordinate wise optimization algorithms
OP1 OP2 OP3	The one-dimensional search in the selected direction
KR1 KR2 KR3	Correction of matrix of variable substitution
ORM	Correction of matrix of variable substitution
EDM	Forming an identity matrix
PAR	Parametric setting of Shore operator
BF1 BF2	The accumulation of information about search directions
DSV	Formation of the search direction using the correction matrix
DG1 DG2	Formation of a diagonal matrix to adjust variable substitution matrix using procedures KR3
CLR	Reset variable substitution matrix
OBR	Extrapolation by two points
OUT	Output of results

Consider the possible algorithms that are used to perform identical tasks.

Installation procedures of algorithmic parameters:

– set the parameters of adaptive local optimization procedures;

– setting wise search algorithms.

Production of random vectors implementations:

– construction of a system of n orthogonal vectors and then transfer to the starting point or the current point of the search;

– interpretation of the realizations of random vectors as the vertices of a regular polygon in space R^n;

– obtaining realizations of random vectors uniformly distributed on the surface of a sphere of radius 1;

– obtaining realizations of random vectors distributed on the surface of the ellipsoid.

Determining the level of the objective function:

– as the level selected “at worst” value of the objective function in accordance with the scheme of Nelder–Mead;

– as the level selected the expectation values of the objective function;

– as the level selected the median of the corresponding distribution.

Check convergence algorithmic procedures:

– stop along the length of the path. The corresponding length of the path is determined by the module DL;

– stop by the standard deviation. The standard deviation is defined by module SIG.

– verification of convergence coordinate wise optimization procedures.

Select search trends in optimization algorithms coordinate wise:

– search is carried out along the coordinate axes of the space of controlled parameters;

– columns in variables replacing the matrix are used as search directions.

Correction of variable substitution matrix:

– adjustment using Shor's statement: when a new search direction (or unpromising perspective), or after a certain number of iterations;

– adjustment by multiplying orthogonal matrix on the diagonal.

The accumulation of information about search directions:

– storage of information about promising research directions;

– storage of information about no promising research directions.

Through various combinations of invariant modules standard deterministic algorithms of search optimization and their adaptive expansion and various modifications can be formed. For example, the module assembly diagram BEG, IN1, SV1, SRT, LV1, P1, M1, DV2, ST, SIG, OS2, OUT matching algorithm alternating polyhedron of Nelder–Mead [2]; scheme BEG, IN2, KD1, EDM, OP1, OS3, OUT – a simple algorithm-wise search; scheme BEG, IN2, KD1, EDM, OP1, OBR, OS3, OUT , obtained from the previous by addition of one module OBR – method configurations by Hook–Jeeves, etc.. Proposed graph structure of algorithmic framework also provides the formation of hybrid algorithmic schemes based on a combination of search optimization procedures for different classes.

VII. CONCLUSIONS

Considered in this paper forming technology of multimethod environment for optimal design leads to the following conclusions. In order to ensure flexibility and adaptability of the algorithmic framework is proposed to use a modular component-based approach to structuring optimization problems and algorithms for solving them. This algorithmic software in general is defined as a set of libraries and optimal selection procedures intelligent interconnecting modules depending on the specific task at hand. The structure of the optimal choice of modules, libraries, characterized by the presence of groups of interchangeable components and the possibility of the formation of different purposes and content of algorithms through modules combinations. Library includes modules of two classes: invariant modules that implement scalar unconstrained optimization procedures; external modules that are connected to the invariant algorithmic kernel depending on the specifics of optimization problem to be solved (dimensions, types of criteria and restrictions, etc.). Represented a generalized model of the library in the form of aggregated graph.

REFERENCES

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

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

[3] G. Berezhnoj, Problems building large IT systems, PCworld, 1998.

[4] A. V. Alekseiev and A. N. Borisov, Intelligent decision system for design solutions. Riga, 1997, 320 p.

[5] V. M. Sineglazov and A. P. Godny, "Dynamic data integration in the design of complex computer-aided design systems", *Electronics and Control Systems,* no. 2(40), 2014, pp. 51–58.

Received January 26, 2016

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: 5.
E-mail: andrewgodny@gmail.com

В. М. Синєглазов, А. П. Годний. Динамічна інтеграція даних під час проектування САПР

Представлено технологію мультиметодного середовища оптимального проектування. Для забезпечення гнучкості і адаптивності алгоритмічного основи пропонується використовувати модульний підхід на основі компонентів для задач оптимізації, структурування алгоритмів для їх вирішення. Алгоритмічне програмне забезпечення в цілому визначається як набір бібліотек і оптимальних процедур відбору інтелектуальних модулів, що з'єднуються в залежності від конкретного завдання.

Ключові слова: динамічна інтеграція, САПР, інтегроване середовище, проектування, оптимальне проектування, алгоритмічний модуль.

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. Аспірант.
Освіта: Національний технічний університет України «Київський Політехнічний Інститут», Україна (2014).
Напрямок наукової діяльності: ідентифікація систем управління, цифрові системи управління
Кількість публікації: 5.
E-mail: andrewgodny@gmail.com

В. М. Синеглазов, А. П. Годний. Динамическая интеграция данных при проектировании САПР

Представлена технология мультиметодной среды оптимального проектирования. Для обеспечения гибкости и адаптивности алгоритмического основы предлагается использовать модульный подход на основе компонентов для задач оптимизации, структуризация алгоритмов для их решения. Это алгоритмическое программное обеспечение в целом определяется как набор библиотек и оптимальных процедур отбора интеллектуальных модулей, соединяющися в зависимости от конкретной задачи.

Ключевые слова: динамическая интеграция, САПР, интегрированная среда, проектирование, оптимальное проектирование, алгоритмический модуль.

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. Аспирант.
Образование: Национальный технический университет Украины «Киевский Политехнический Институт» (2014).
Направление научной деятельности: системы и процессы управления, идентификация, автоматизация систем управления технологическими процессами.
Количество публикаций: 5.
E-mail: andrewgodny@gmail.com