Chapter

The Technique of Automated Design of Technological Objects with the Application of Artificial Intelligence Elements

Tatyana Zubkova and Marina Tokareva

Abstract

The chapter describes the methodology of using artificial intelligence methods to build an integrated environment for computer-aided design components of technological objects based on their classification, integration and configuration. It describes the formation of CAD based on the object-oriented approach, methods of configuring the integrated environment and the organization of single information space. The configuration of the system components and the methodology for organizing the interaction of CAD components, obtaining the final CAD architecture, focused on solving the problem, is shown. The application of the Mamdani method for the formal description of project operations and the use of genetic algorithms to optimize the operational parameters of the process and the design of the technological machine are described.

Keywords: technological machines, CAD, object-oriented approach, information system, configuration of components, integration of components, creation of a single information space, fuzzy inference algorithms, genetic algorithm

1. Introduction

Technological machines have a fairly wide range of applications in all areas of person's production activity. Designing them, it is necessary to take into account the properties of the material being processed, the requirements for the technological process, the quality of the finished product, as well as the geometric and design features of the machine itself. Market competition forces manufacturers to improve and create new technologies to increase the range of their products. Therefore, production must be flexible, with the ability to re-adjust to different types of raw materials, product configurations, productivity, etc., depending on the current market needs.

The design of sophisticated technological machines is currently focused on the use of computer-aided design (CAD) systems to solve a wide range of engineering problems: strength calculation, dynamics, kinematics, heat transfer, acoustics, durability, etc., modeling of technological processes of manufacturing and product assembly. This is achieved by combining modern hardware and software, the parameters and characteristics of which are selected with maximum consideration for the features of the tasks of the design process.

The purpose of the work is to show the methodology of computer-aided design of technological objects using intelligent methods. The volume of design and engineering work, the proposed approaches, allows to increase the productivity of design engineers. The use of artificial intelligence methods is particularly relevant when there are no well-developed methods of designing or creating a fundamentally new and demanding creative work.

2. Statement of the problem of CAD formation based on an object-oriented approach

An adequate information system (IS) is necessary for designing technological objects.

A feature of solving the problems of designing technological objects is the presence of specialized subsystems. The need for a narrow specialization of the components of the applied area is justified by the following factors:

- ease of component data abstraction for the users and for other software components;
- the possibility of parallel development of components of a specific task;
- modification and replacement of the component with an alternative one, if the equipment or the target platform has changed, without changing the other components.

The requirements presented can be achieved by applying object-oriented decomposition. Based on this decomposition for the designed CAD components, the technological requirements may be as follows:

- components must have certain software interfaces for interaction;
- for the implementation of components it is necessary to use the principles of object-oriented design and programming;
- component functions must solve certain tasks [1–4].

The object-oriented approach allows us to consider the CAD system as an independent system S, the properties of which are inherited from the components included in its composition. The system S has a finite number of characteristics $F = \{F_i\}, i = \overline{1, n}$, where n is the number of properties. Let it be m possible ways of forming the system S. In the $k - y$ $(k \in m)$ method of formation $S_k = \{R_p\}, p = \overline{1, P_k}$, where P_k is the number of subsystems in the $S\ k - y$ decomposition method, each CAD resource (component) is characterized by a set of properties $F = \{F_{P_k}\}, k = \overline{1, K}$, each of which has an individual numerical measure. The set of properties of all resources R_k in $k - y$ the first decomposition $F_k = \cup_{p=1}^{P_k} \{F_{P_k}\}$. Interacting with each other, resources generate many system processes $Z_k = \{Z_{kj}\}, j = \overline{1, J}$, where J is the number of processes.

DOI: http://dx.doi.org/10.5772/intechopen.88295

The optimal organization of CAD is the selection and distribution of resources $r \in R_k$ between project tasks Z_k according to a given decomposition scheme k to ensure the extreme values of the system properties $extr(F_k)$ necessary to perform the required operations. At the same time, the time t_k to solve each of the project tasks Z_k while ensuring $extr(F_k)$ should be minimized (**Figure 1**).

Formalization of the selected conditions is represented as a system (1).

$$\begin{cases} \sum_{j=1}^{J} Q_j \left(\bigcup_{p=1}^{P_j} \{F_{jp}\} \right) \to \max; \\ \sum_{j=1}^{J} t_j \left(\bigcup_{p=1}^{P_j} \{F_{jp}\} \right) \to \min. \end{cases} \tag{1}$$

where Q_j is the selection function of system properties, under which the quality characteristics for each design task are maximized; t_j is the selection function of system properties, which minimizes time for each project task; F—finite set of characteristics; P—number of subsystems; J—number of processes.

System (1) contains two particular criteria with different directions of optimization (for qualitative characteristics, maximization, for temporal characteristics, minimization).

To determine the target function we use the additive criterion. The objective function is formed by adding the normalized values of the partial criteria and in the case of the application of the additive criterion will take the form:

$$\sum_{j=1}^{J} \left(c_j \frac{Q_j(F_j)}{Q^0{}_j} - v_j \frac{t_j(F_j)}{t^0{}_j} \right) \to \max, \tag{2}$$

where F_j is the property of the alternative subsystem for solving the j-y design problem (controlled parameter), and $F_j \in \cup_{p=1}^{P_j} \{F_{jp}\}$, $Q^0{}_j$, $t^0{}_j$—j-y normalizing divider for quality and time characteristics, respectively; c_j v_j—weights of the j-y particular criterion.

When searching for the values of the objective function, the qualitative characteristics have a higher priority; if the values of the qualitative indicators are equal (the presence of the required properties) of the program components, the choice is made according to the time characteristics. Thus, the values of the weighting factors must meet the condition $\frac{c_j}{v_j} \to \infty$.

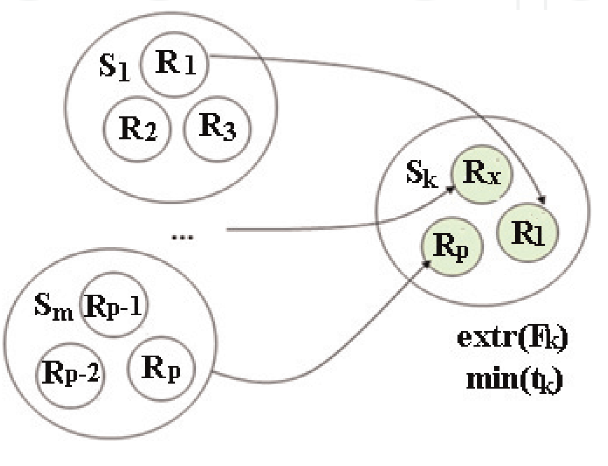

Figure 1.
Organization and distribution of resources between project tasks.

The controlled parameters are the properties of the system (F) and the totality of the components (R) that provide these properties:

$$U = \begin{cases} F, \\ R. \end{cases} \tag{3}$$

In real conditions, the choice of the values of the controlled variables, most often, is imposed by the limitations associated with the available resources, power and other features. Functional limitations establish certain dependencies between the controlled parameters, which cannot be violated under the terms of ensuring the performance or efficiency of the technical system [1, 5].

The development of an integrated environment by adding new software modules that automate individual functional and managerial procedures makes it necessary to create a methodology for managing the configuration of an integrated environment.

3. Components of the methodology for configuring the integrated environment and the organization of a single information space

In relation to the problem area of development, the following components of the methodology are highlighted (**Figure 2**):

- single realizable environment;
- classification of system components;
- integration interfaces;
- integration principles of components;
- configuration of system components.

Classification of system components form a generalized representation of groups and subgroups of components of a software system (PS).

A single executable environment is a software environment where components function and interact with each other.

Integration interfaces are software interfaces that provide interaction between software systems of a single executable environment and implemented components.

Component integration principles—a set of rules necessary for the interaction of two or more components. Since complex CAD consists of components that perform

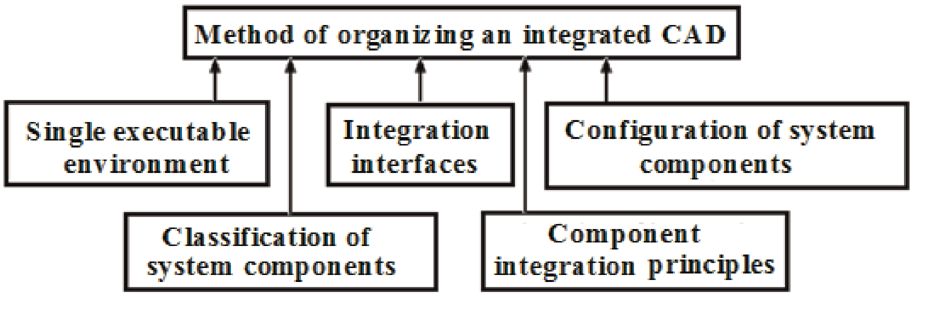

Figure 2.
Components of integrated CAD.

synthesizing, analyzing, evaluating, converting functions, it is necessary to consider various principles of the organization of interaction between subsystems.

Configuration of system components is a change in the set of software components depending on the design tasks to be solved [6, 7].

The use of various automated systems (AS) by independent developers in the product life cycle processes raises the problem of the information compatibility of these systems, which limits the possibility of using the same data and sharing it. This necessitates the presentation of data in the form of a single structured information model that is accessible to all specialists in the design process. The common information space (UIS) allows:

- accept and store the product design electronically;
- keep track of the current state of the product;
- organize a quick view of all models and documents;
- to ensure the rapid exchange of information between users of the integrated environment;
- to ensure information consistency and exchange between all subsystems of CAD.

These requirements for the UIS can be fulfilled if the synthesizing, constructive and analyzing design processes in the CAD system are automated. And also, if the project information enters the information space automatically and is available to all users of the system in accordance with the existing access rights.

In most cases, the integration of various CAD subsystems is carried out using API (application programming interface) interfaces (**Figure 3**) and CAD (computer aided design), CAE (computer aided engineering) systems.

The disadvantages of organizing this interaction are:

- narrow specialization of the developed integration interface designed to solve the interaction problem only for certain subsystems;
- dependence of the functioning of the system on changes in the API;
- the need to develop additional integration when introducing new CAD and CAE systems;
- lack of free access to API software products.

Thus, the integration option, based only on the API, allows you to create the core of the enterprise UIS, but in this case the system is limited from changes in API functions to changes in the level of replacement of subsystems.

A fundamental solution to the problems considered is the introduction of CALS-technologies. CALS (continuous acquisition and lifecycle support—continuous information support for supplies and product life cycle) is a modern approach to the

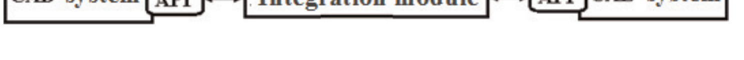

Figure 3.
Interaction scheme of CAD-system and PS through API.

design and manufacture of high-tech and knowledge-intensive products, which consists in using computer technology and modern information technologies at all stages of the product life cycle. The main CALS tool is SDE (shared data environment).

The CALS approach is to free the user from dependence on the manufacturer of the software being used. The basis of the approach is a single information space in accordance with the international standard for data presentation. The main standard is ISO 10303 STEP (standard for exchange of product model data), based on the express language. There are also standards IGES (international graphical exchange format) and DXF (drawing interchange format), which are currently one of the main data storage formats for some CAD-systems. As an alternative language for the exchange of geometric and technical data about the product can be used XML markup language. For CALS, of interest are the subsets of product definition exchange (PDX) and 3D XML dedicated to data exchange in CAD.

Due to the considered features, the general organization of interaction between the CAD subsystems looks like that shown in **Figure 4**. The use of CALS technologies can significantly reduce the scope of design work, since the descriptions of machines and systems, many components of equipment that were previously designed, will be stored in the PS database in unified format.

The solution of the integration task is to integrate ready-made software solutions using interfaces into a single system of modules. The connecting element is a special module designed for the transmission and transformation of data. Formed a single executable environment provides a functional basis for the introduction of various components into the system [8–10].

The interaction of integration components with external CAD systems (CAD and CAE systems) and modules of the PS being developed is presented in **Figure 5**.

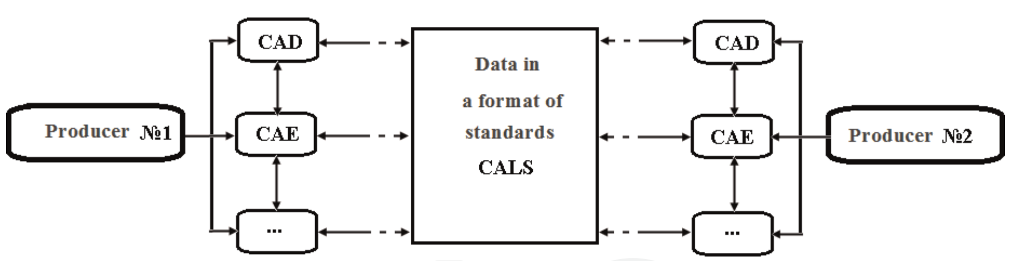

Figure 4.
CALS technology.

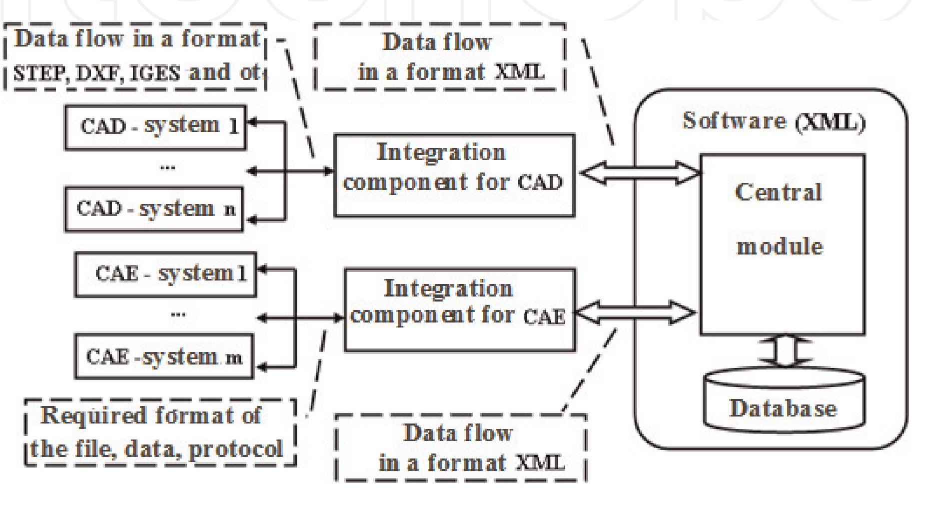

Figure 5.
The interaction of systems in the integrating complex.

CALS standards have been chosen as the format for exchanging design data. Thus, according to the presented scheme, geometric models presented in STEP, DXF, IGES formats can be used as imported (exported) design information. The format of data exchange between the internal components of the software core (PS) is the product model, represented in the XML markup language. The use of a similar organization of the internal representation of the central module will allow to include in the product model, in addition to the design parameters, also the parameters of the technological process, the rheology of the material, temperature characteristics in different parts of the machine and other characteristics necessary for analyzing the designed structure and its identification in the system.

The interaction of external components of the integrated environment is provided by:

- using exchange files;
- unidirectional or bidirectional software interfaces;
- using database tables (DB).

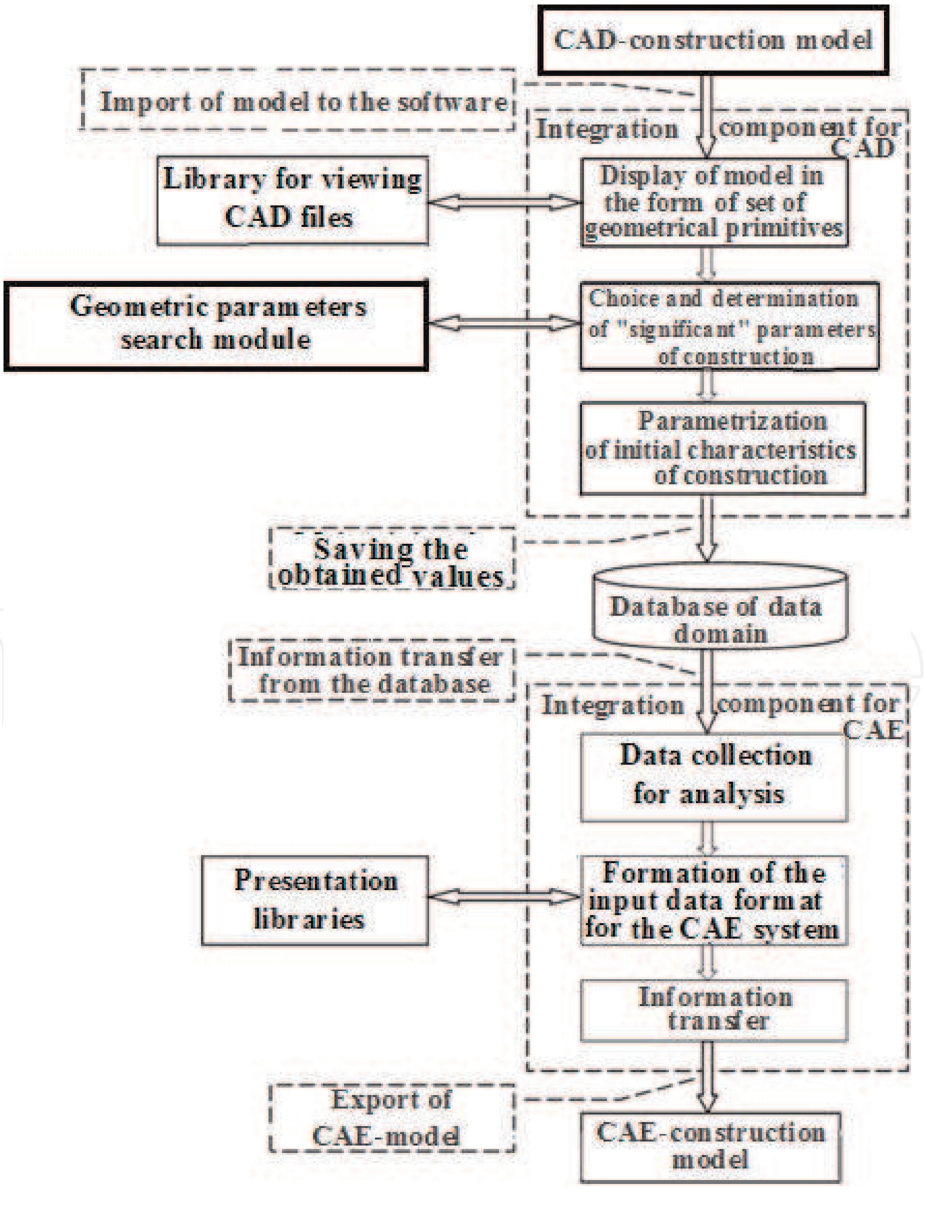

Figure 6.
Methods of organizing the interaction of CAD and CAE-systems.

The use of the software interface provided by the manufacturers restricts the functionality of the system, its extensibility and scale. Thus, exchange files are selected as the main method of interaction between external CAD components. The connection between the components of the central module is organized by building a database of the subject area.

CAD systems for design and CAE systems for analysis interact poorly with each other, despite their wide distribution. CAD and CAE models use different types of geometric models, and there is no common unified model that contains information for design and analysis.

As a solution to these problems, a variant of CAD-CAE bidirectional integration is proposed. Then the system will allow the CAD system to automatically generate models for analysis, and the CAE system will automatically modify the geometry of the parts and carry out a new analysis. The transformation process will be repeated until the specified criterion is reached.

Based on this approach, a method was developed for obtaining the required CAE model from an imported structural drawing (**Figure 6**).

The technique is implemented by a sequence of steps performed in the integration components. Each of the integration stages is implemented as a separate functional unit.

According to the CAD/CAE-integrated approach, the implementation of the inverse transform from the CAE model to the geometric representation of the structure is also required. In this case, the technique will be similar to that shown in **Figure 6**. At the same time, the functional blocks will also be located in the integration components, which will significantly reduce the complexity of implementing the interaction between the CAD and CAE subsystems.

Based on the bi-directionality and versatility of this technique, it is possible to track product adjustments at each of the design stages. This will automatically modify the CAD/CAE model in accordance with the changes made.

4. Configuring system components and methods of organizing the interaction of CAD components

In CAD, models are presented in the form of problem-solving algorithms, and then in the form of software. Technological objects, divided into private submodels, are divided into simpler individual aspects of the object's functioning (that is, they are decomposed into particular models). Each individual model is represented by some mathematical transformation (**Figure 7**).

Figure 7—$Z = \{z_i, i = 1, \dots, k\}$ is a set of output parameters of the model; V—operator (model) of the transformation (V—function of the input variables); A vector $X = \{x_i, i = 1..n\}$ is a set of external parameters coming from a model of a more general system; A vector $Y = \{y_i, i = 1, ..m\}$ is a set of input controlled

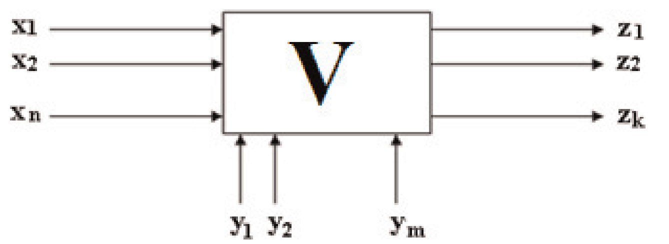

Figure 7.
Mathematical transformation.

DOI: http://dx.doi.org/10.5772/intechopen.88295

parameters of the model that the designer can operate on during the design process. Controlled input parameters can vary within specified limits, i.e., so-called parametric constraints are imposed on them: $y_i^н \le y_i \le y_i^в, i = 1..m,$ where $y_i^н$ and $y_i^в$ are lower and upper limits.

With regard to the technological machine, the requirements for the technological process, which affect the kinematic parameters, the structural parameters of the mechanism and the geometrical parameters of the interaction space, act as the rheological properties of the material being processed as input parameters of the mathematical model (MM). All this affects the internal characteristics of the system, which determines the scale, efficiency of the process and the quality of products.

Building techniques and methodological principles for the operation of complex systems simplifies and streamlines the process of developing CAD software, and presents a sequence of operations in the form of a logic diagram. This methodology represents an approach to the design and development of CAD, based on the interaction of software solutions available on the market in the field of automation and its own software design algorithms.

Figure 8 shows the sequence of steps in the proposed method of functioning CAD systems.

In the *first step* of the implementation of the methodology, tasks (including new ones) and a set of components for their implementation are determined based on the initial design requirements. The choice of a set of components is implemented according to their classification by functionality. *Next* is the organization of the interaction, configuration, and the formation and exchange of data. Thus, the final set of components will allow you to find solutions to all the tasks, or to determine the set of tasks that cannot be solved with the existing set of subsystems.

After performing the first step, we obtain a specific set of components and an approximate system architecture. Each component performs a specific task at a certain level of logical implementation.

In the *second step*, a single executable environment is formed.

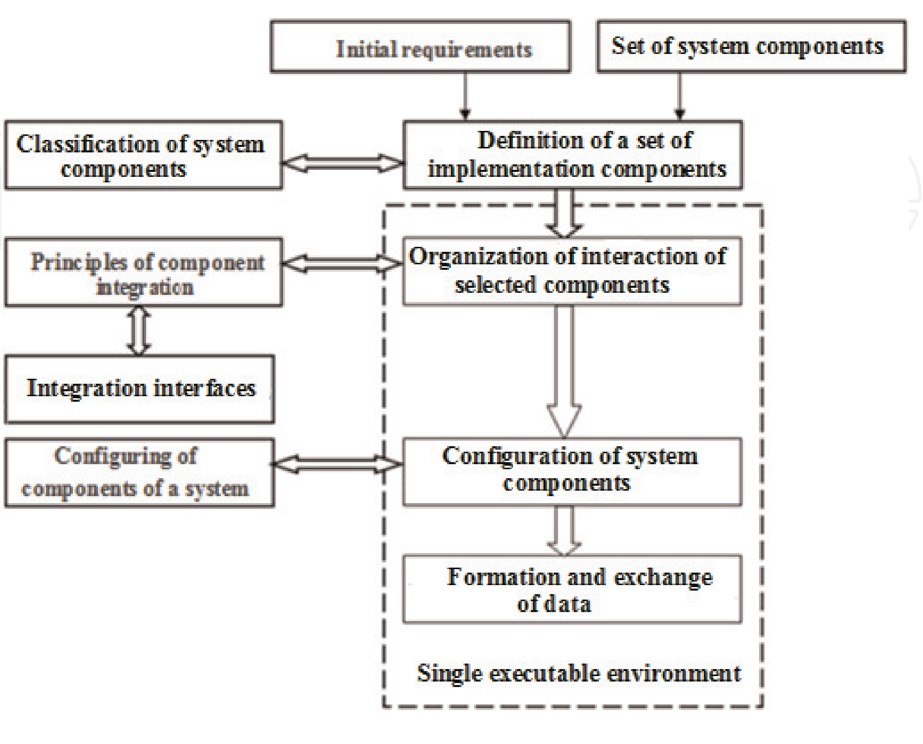

Figure 8.
Technique of functioning of CAD.

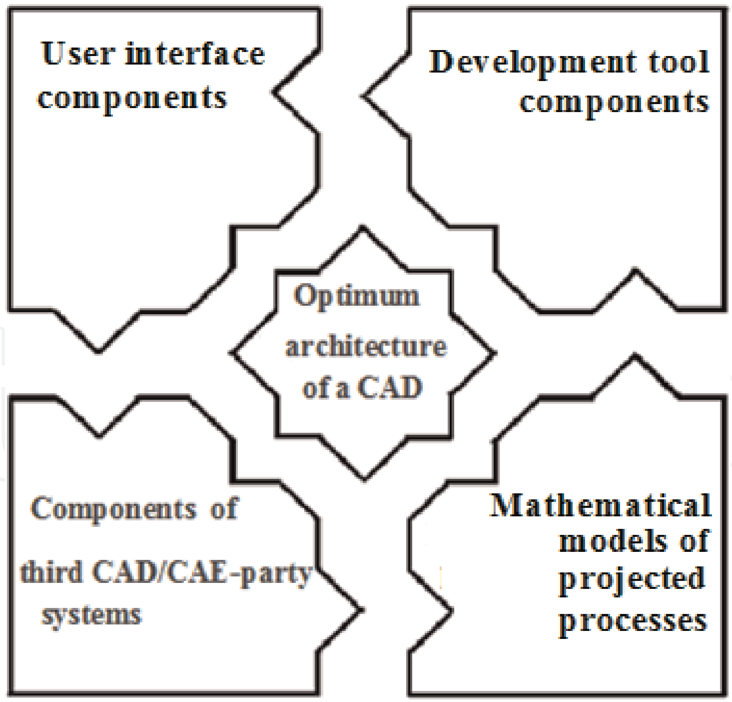

Figure 9.
Configuring CAD components.

The *first stage* of formation is the realization of interaction between the selected components. The interaction is carried out in accordance with the principles based on the use of integration and implementation interfaces and a connecting component. The interfaces required for the operation of each component are determined.

The *second stage* is the configuration of the selected components. This stage involves the selection among alternative algorithms that involve solving the same tasks, the most adapted for the design of a particular product (initial system requirements) [11, 12].

Further, in the *third stage* of the second step, the source data is modified to exchange them between the selected components. An example would be a change in the geometric design in the product model required for analysis.

The result of the *second step* is a formed single executable system environment, which has mechanisms for implementing external PSs and for ensuring their interaction with the components of the entire system. This is the final architecture of CAD, focused on solving the problem (**Figure 9**).

5. Formal description of project operations

Supporting the choice of the right MM configuration among many alternatives is similar to the task of building CAD systems, which consists in allocating resources R_k between design tasks Z_k in such a way as to ensure optimal values of system properties $extr(F_k)$ in the system S_k (**Figure 1**).

To apply each of the alternative mathematical models, it is required to check the ability of its functioning in a given configuration of the system. The graph of component configuration is supplemented with transition events (U_k) and is represented as a Petri net for the system S_k method of decomposition (**Figure 10**) [13].

As conditions for the transition (U) to a specific MM, we will use verification of the required set of initial data, including constructive (U_K), geometric (U_G), kinematic (U_P) and rheological (U_R) parameters.

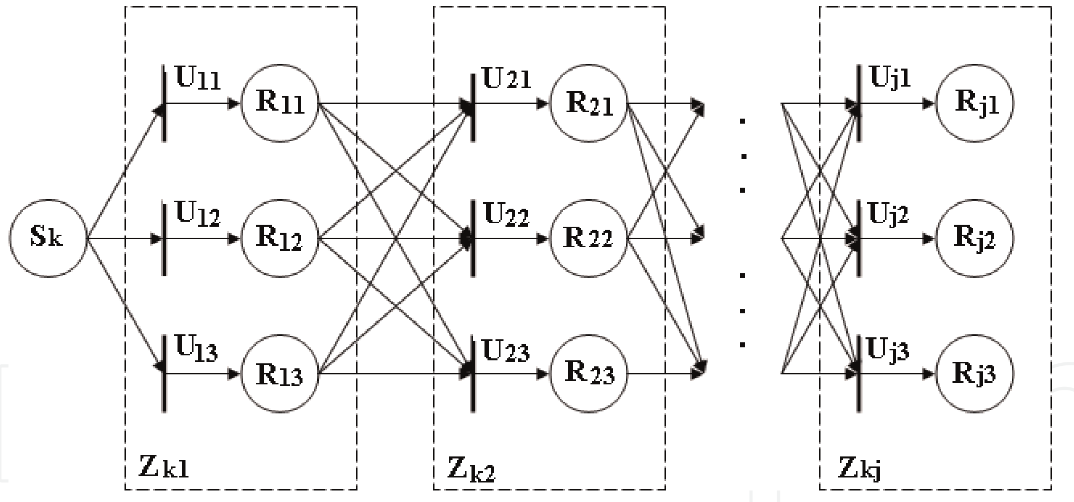

Figure 10.
MM component configuration graph.

Thus, the system S_k must solve many design problems $Z_k = \{Z_{kj}\}, j = \overline{1, J}$, J is the number of tasks. To solve each project problem there is a set of alternative resources (components) $R_j = \{R_{jy}\}, y = \overline{1, Y}$, Y—the number of resources intended for solving the problem. Each resource R_{jy} is characterized by a set of properties $F_{jy} = \{F_{jyl}\}, l = \overline{1, L}$, L is a set of properties of the j resource. The performance of each of the properties of the resource is limited by the condition of transition to the specified resource $U_j = \{U_{jy}\}, y = \overline{1, Y}$.

The task of optimal configuration of MM is to allocate resources $r \in R_j$ between project tasks Z_k in such a way as to provide extreme values of system properties, as well as to support the ability to use each of the selected resources $extr(F_k) \wedge M_j(U_j,\ R_j,\ Z_k,\ S_k)$ (where M_j is the function of checking transition U_j to a resource R_j to solve the problem Z_k in the system S_k).

Thus, the objective function of building CAD (2) is relevant for the configuration of MM. At the same time, the system of control parameters will not change its composition (3), the system of restrictions, which is characterized by design tasks, design features of the product, technological requirements of production will be complemented by transition conditions (U):

$$C = \begin{cases} Z, \ \ where \ Z \in \{Z_S,\ Z_A,\ Z_E,\ Z_C,\ Z_V,\ Z_D\}, \\ K, \ \ where \ K \subseteq \{\ K_I,\ K_Z,\ K_S,\ K_F,\ K_{FI},\ K_O,\ K_P,\ K_{FS}\}, \\ T, \ \ where \ T \subseteq \{\ T_n,\ T_\mu,\ T_\sigma,\ T_t\}, \\ U, \ \ where \ U \subseteq \{\ U_K,\ U_G,\ U_P,\ U_R\}. \end{cases}$$

But it is impossible to apply optimization tasks to the described subject domain based on graph theory, since there is no complete information about the relationships between the components:

- management;
- according to information;
- by placement;
- by effects.

Decision making in most cases consists in generating possible alternative solutions, evaluating them and choosing the best option. When choosing an option, one has to take into account a large number of uncertain and contradictory factors. Uncertainty is an integral part of decision-making processes [14–16].

To do this, use systems based on "soft" calculations, which include:

- probabilistic calculations and fuzzy logic;
- neurocomputing—training, adaptation, classification, system modeling and identification;
- genetic computation—synthesis, tuning, and optimization using systematic random search and evolution.

Fuzzy inference algorithms mainly differ in the type of rules used, logical operations, and the type of dephasing method. There are models of Mamdani, Sugeno, Larsen, Tsukamoto.

The Mamdani method is the most common method of inference in fuzzy systems. It uses the minimax composition of fuzzy sets. This method includes the following sequence of actions in relation to the task of reconfiguring MM:

1. Formation of the rule base. The rules are as follows: If <condition 1> and <condition 2> ... and <condition n>, then <output>. The conditions indicate the compliance of the input parameters X_i ($i \in [1, \ldots, n]$) to the requirements. Based on the input parameters, as well as the estimated opinion, <condition> takes a value in the interval [0 ... 1]. "Inference" corresponds to the choice of using the component for which the rule is made.

2. Fuzzifikation of input variables. This stage is called reduction to illegibility. The input contains the generated rule base and the input data array $A = \{a_1, \ldots, a_m\}$, where m is the number of input variables. The purpose of this stage is to obtain truth values for all sub-conditions from the rule base. For each of the sub-conditions, there is a value $b_i = \lambda_i(a_j)$, where λ is the membership function, which associates the specific values of the degree of truth with all values of the input variables; j = 1,..., m; i = 1,..., k, where k is the total number of sub-conditions in the rule base. Thus, a set of values is obtained b_i.

3. Aggregation of sub-conditions. The purpose of this stage is to determine the degree of truth of the conditions for each rule of the fuzzy inference system: $c_i = \min\{b_i\}$.

4. Activation of sub-conclusions. At this stage there is a transition from the conditions to the sub-conclusions. For each subconclusions is the degree of truth $d_i = c_i \cdot F_i$, where $i = 1, \ldots, q$, q—the total number of subconclusions in the rule base, F—weights, meaning the degree of confidence in the truth of the subconclusions. Then, again, i for each subconclusion, the set is compared D_i with the new membership function. Its value is determined as a minimum from d_i and the values of the component membership function from the sub-clauses. This method is called min-activation, which is formally written as follows: $\lambda_i'(x) = \min\{d_i, \lambda_i(x)\}$.

5. Accumulation of conclusions. The purpose of this stage is to obtain a fuzzy set (or their union) for each of the output variables. It is executed as follows: i

output variable is associated with the union of the sets $E_i = \cup D_j$. Where j—numbers of sub-conclusions in which the output i variable participates $i = 1..s$. The union of two fuzzy sets is the third fuzzy set with the following membership function: $\lambda_i^{'}(x) = \max\{\lambda_1(x), \lambda_2(x)\}$, where $\lambda_1(x),\ \lambda_2(x)$ are the membership functions of the joined sets.

6. Defusing output variables. The purpose of defuzzification is to obtain a quantitative value (crisp value) for each of the output linguistic variables. The i output variable and the related set are considered. Then, using the defuzzification method, the total quantitative value of the output variable is found $E_i(i = 1..s)$. In this algorithm implementation, the center of gravity method is used, in which the value of the i output variable is calculated using the formula:

$$y_i = \frac{\int_{\text{Min}}^{\text{Max}} x \cdot \lambda_i(x)dx}{\int_{\text{Min}}^{\text{Max}} \lambda_i(x)dx},$$

where $\lambda_i(x)$—membership function of the corresponding fuzzy set E_i; *Min* and *Max*—boundaries of the universe of fuzzy variables; y_i—result of defuzzification.

The advantage of the method is the ability to take into account the unlimited number of various conditions and make the rules of various forms. The accuracy of the results depends on the size of the knowledge base.

Thus, based on the result of the defuzzification stage for a specific rule, one can judge about the need to use a specific MM.

6. Optimization of process parameters and technological machine design

On the basis of the decomposition of the technological process and its mathematical model, a reverse action can be carried out—the compositional design of the technological object. The method of composite design allows to achieve the optimal design solution [7].

As a rule, when searching for the optimal combination of design parameters of a technological machine, it is necessary to control its thermal and mechanical tension, also the intensity of interaction of the material with the working parts of the pressing mechanism and other control factors that limit the area of optimal search and are limitations. In view of the presence of constraints on design optimization, without the possibility of simplification and omission, it reduces the current type of optimization to the problem of conditional multiparameter optimization.

Among the methods of conditional multiparameter optimization, there are three approaches. The first is based on the distribution of criteria by importance and their consistent optimization with allowance for the tolerance. The purpose of the second approach is to single out the main criterion and translate the rest into restrictions. The third approach is the convolution of a vector criterion into one generalized criterion. However, this method is very inconvenient, since to find all effective points (Pareto sets), it is required to change the coefficients of the parameterized function of the global criterion. The second drawback is the need to repeat the algorithm several times (the number of iterations is equal to the power of the Pareto set) to obtain the entire set of effective points.

Thus, the considered classical methods have drawbacks when solving multicriteria optimization problems. It seems more promising to use an evolutionary approach to speed up the process of obtaining Pareto-optimal points, thanks to the ability of genetic algorithms (GA) to find a solution even when the algorithm is executed once.

GA do not guarantee that a global solution will be found, however, they are good for finding a "good enough" problem solution "fast enough." Even where existing techniques work well, improvements can be achieved by combining them with GA.

The genetic algorithm works with a certain objective function $Q(u_1, u_2, \ldots, u_n)$ and as a result finds either its maximum or minimum (depending on the task). It does not require finding the derivative of the function Q and other calculations, since the genetic algorithm considers the objective function as a block (a set of some actions, operations and calculations), which at the input receives a certain set of values $u_1, u_2, \ldots, u_n, u_1, u_2, \ldots, u_n$ and the output gives the result, directly dependent on them.

The GA, which includes the ability to set various selection criteria, will allow to take into account the boundary conditions necessary for the successful course of the technological process.

It should be noted that the use of classical (exact) mathematical optimization methods is not always appropriate, because The simulation model is not an absolute copy of the real system (there is a certain degree of accuracy), while the use of exact methods requires significant computational costs, which is critical in many cases. Therefore, as a search engine optimization algorithm, it is more expedient to use a method that does not necessarily guarantee the achievement of an exact optimum, but finds solutions that are close to optimal, and at the same time ensures fast search convergence of the algorithm.

The work of a genetic algorithm is an iterative process that continues until a specified number of generations or some other stopping criterion is fulfilled. A generalized block diagram of the work of the GA is shown in **Figure 11**.

Of particular importance is the creation of the initial population. The rate of convergence of the method and the place of the local maximum depend on the choice of the initial data. Often this choice is made randomly, but it is rational to use the results obtained earlier, which are contained in the database.

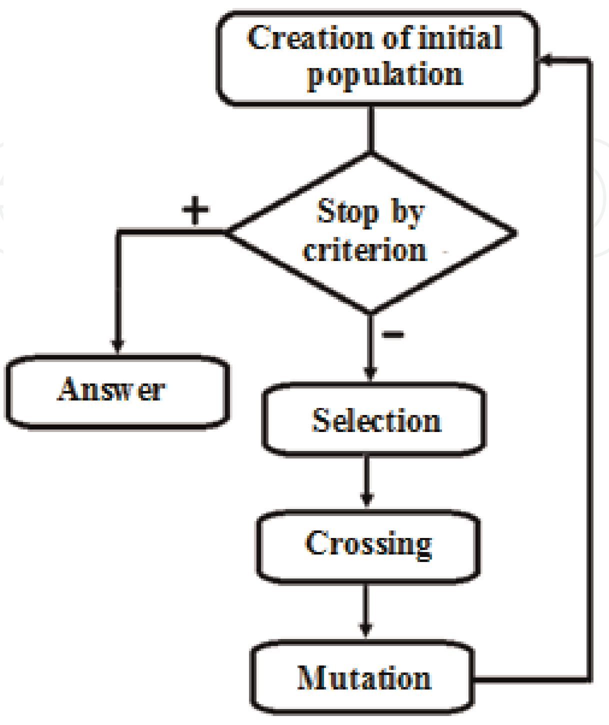

Figure 11.
Generic genetic algorithm scheme.

As a method of selecting values from a set of results, taking into account the criteria and objectives of optimization imposed on them, we will use the Mamdani method (Section 4), based on "soft" calculations.

It uses a minimax composition of fuzzy sets:

$$MF = \max(\min(A_{ik}(x_k)))$$

where x_k—input variables (extruder design parameters D_1, $s_ш$, $p_ш$, $h_ш$, L); A_{ik}—given fuzzy sets with membership functions of input variables to the required parameters.

At each iteration of the GA, selection, single-point crossover and mutation are implemented.

At the selection stage, the "fitness" of each structure is evaluated. The selection is made by assigning each structure a probability equal to the ratio of its adaptability to the total adaptability of the population:

$$P_S(i) = \frac{f(i)}{\sum_{j=1}^{N} f(j)},$$

where $i = 1, \ldots, N;\ N$—number of estimated machine designs; $f(i)$—compliance with the requirements i—construction.

In genetic algorithms, a crossover operator (crossover) is responsible for transferring descendants of parents to descendants. In the simplest case, the crossover in the genetic algorithm is implemented as shown in **Figure 12**.

Since in this case the information about the construction is presented in the form of a set of real numbers, therefore, a type of genetic algorithm is used, which is called continuous GA or a genetic algorithm with real coding.

A continuous GA crossing operator generates one or more descendants from two parent sets. As a matter of fact, it is required to get new vectors from two vectors of real numbers according to some laws. Most of these algorithms generate new vectors in the neighborhood of parent pairs.

In this case, an extended line crossover is considered (**Figure 13**), which can formally be represented as a formula:

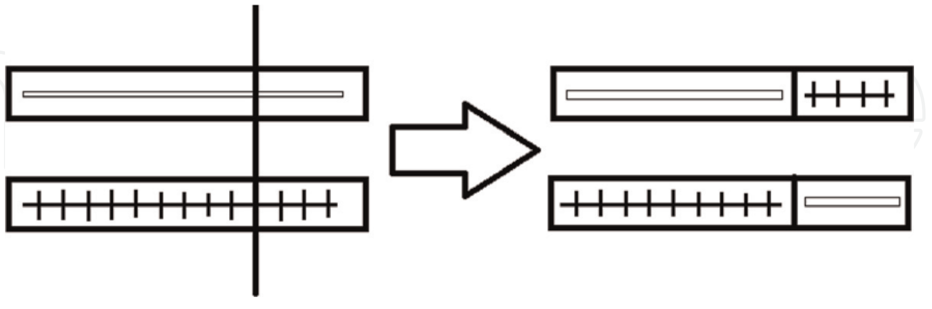

Figure 12.
Concept of crossing over.

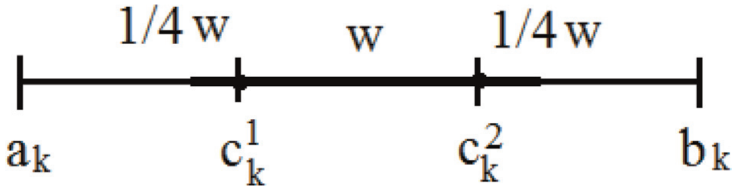

Figure 13.
Line crossover.

$$h_k = c_k^1 + w \times \left(c_k^2 - c_k^1\right)$$

where h_k–k—construction parameter; $H = (h_1, \ldots, h_n),\ \ k = \overline{1, n}$; n—number of parameters; c_k^m–k—parent element $C^m = \left(h_1^m, \ldots, h_n^m\right),\ \ k = \overline{1, n},\ \ m = \overline{1, 2}$; w—random number of interval $[-0, 25;\ \ 1, 25]$.

A mutation is a random change in one or more positions in a set of characteristics, designed to maintain diversity and to protect against premature convergence.

The mutation element will be defined as follows:

$$x_k = a_k + (b_k - a_k) \times u,$$

where x_k–k—construction parameter $X = (x_1, \ldots, x_n),\ \ k = \overline{1, n}$ determined on interval $[a_k;\ \ b_k][a_k;\ \ b_k]$; u—random number of interval $[0;\ \ 1][0;\ \ 1]$.

Crossing procedures, mutations, assessment of fitness, the choice of the best solution are repeated in the cycle until the stop criterion works. The fulfillment of a certain condition is used as such a criterion. The essence of the condition is as follows: if the difference between the best solutions at the moment and at the previous iteration is insignificant (less than a certain set ε), or the maximum number of iterations is completed (the number of iterations performed is more than a certain set n), the algorithm stops its work. The last best solution found is the optimal set of parameters.

This method has several disadvantages—it is convergence to a local optimum, lack of accuracy of the results.

Therefore, if it is necessary to optimize a specific parameter when changing only one indicator and the invariance of others, then it is more rational to use exact methods of multidimensional optimization (the coordinatewise descent method, the Hook-Jeeves method, etc.).

The advantage of applying this modification of the genetic algorithm is the scalability of the optimization problem. Using this method, it is possible to optimize a different set of parameters, as well as to establish all possible optimization conditions, both for a single resulting indicator and for several. The method does not require significant computational costs, which are necessary for the implementation of exact methods, and which are often impossible in complex systems. GA provides fast search convergence of the algorithm. The results of the method satisfy the conditions of the technological process, as obtained by simulation.

As a solution to the identified deficiencies, it is possible to combine the presented method with accurate methods of multidimensional optimization.

7. Conclusion

The implementation of the developed methodology has several advantages for developers and end users:

- costs are reduced for the development of user interface and application applications (for example, control program editors, network connection configurators, etc.);

- it is possible to determine the required set of components, and the tasks they implement, in the early stages of designing applied software;

- reduced time of release of the new system, due to the possibility of issuing a lightweight (preliminary) version with a subsequent increase in functionality.

The developed technique with the use of artificial intelligence methods of building an integrated environment allows organizing the interaction of CAD components based on their classification, integration and configuration.

Author details

Tatyana Zubkova* and Marina Tokareva
Orenburg State University, Orenburg, Russia

*Address all correspondence to: bars87@mail.ru

References

[1] Mustyukov NA, Zubkova TM. Application of the genetic algorithm for carrying out parametrical synthesis of construction of an extruder. 2013;**4**:114-118. the ITMO Scientific and technical bulletin St.Petersburg State University

[2] Korobeynikov AG, Fedosovsky ME, Zharinov IO, Shukalov AV, Gurjanov AV. Development of conceptual modeling method to solve the tasks of computeraided design of difficult technical complexes on the basis of category theory. International Journal of Applied Engineering Research. 2017;**12** (6):1114-1122

[3] Gorbunov AA, Pripadchev AD, Bykova IS, Elagin VV. Simulation modelling for computer aided design of secondary aerodynamic wing surfaces. Advances in Systems Science and Applications. 2015;**15**(4):346-358

[4] Gorokhov Yu V, Timofeev VN, Belyaev SV, Kirko VI, Avdulov AA, Konstantinov IL, et al. Conform installation structural elements design methods. Journal of Engineering and Applied Sciences. 2017;**12**(9):2917-2922

[5] Zubkova TM. Development of methodology of mathematical modeling of technological objects. 2002;**2**:209-213. Messenger of the Orenburg state university

[6] Zubkova TM. Automated design of the extrusion equipment using intellectual systems. Intelligence. Innovations. Investments. 2017;**8**:51-55

[7] Kirillov NP. Description of the method of combined conceptual modeling of technical systems. Tr. SPIIRAS. 2013;**31**:223-235

[8] Zubkova TM, Mustyukov NA, Tokareva MA. Creation of architecture of a CAD of one-screw extruders using elements of artificial intelligence. 2016; **4**:176-182

[9] Vasiliev PV, Kuzenyi VV. Improving the effectiveness of scenario development in the intellectual-modeling environment. Tr. SPIIRAN. 2011;**19**:288-297

[10] Shishigin DS. To the choice of integration technology for application software with CAD. Proceedings of SPIIRAS. 2016;**4**(47):211-224

[11] Zubkova TM. Automated industrial design based on artificial intelligence. Russian Engineering Research. 2018; **38**(5):394-398

[12] Zubkova TM, Tokareva MA, Sultanov NZ. Creation of system of computer-aided design for technological objects. Journal of Physics: Conference Series. 2018;**1015**:052031. DOI:10.1088/1742-6596/1015/5/052031. Techniques and technologies of the automated mechanical engineering.

[13] Zubkova TM, Mustyukov NA, Kolobov AN. Reconfiguring of a CAD for design of one-screw extruders on the basis of model of an indistinct output of Mamdani. 2013;**1**:176-181. the Messenger of regional public institution

[14] Troitsky AK. Two-level multiple face detection algorithm based on local feature search and structure recognition methods. International Journal of Applied Engineering Research. 2016; **11**(6):4640-4647

[15] Kobersy IS, Shkurkin DV, Zatonskiy AV, Volodina JI, Safyanova TV. Moving objects control under uncertainty. ARPN Journal of Engineering and Applied Sciences. 2016; **11**(5):2830-2834

[16] Molodtsov DA. Soft probability of large deviations. Advances in Systems Science and Applications. 2013;**13**(1): 53-67