MULTIAGENT METHODOLOGY FOR COMPUTER AIDED DESIGN AND INTEGRATED MANUFACTURING

G. B. Evgenev
Bauman Moscow State Technical University, 107005, Moscow, 2-nd Baumanskaya st., 5,Russia tel. (095)263-69-70, fax. (095)263-60-57, E-mail. office @ sprut.bmstu.ru

Abstract
Multiagent methodology is a most promising tool to obtain the integrated intelligent system of design and manufacturing. This paper deals with systemic classification of models of engineering knowledge. A formal design theory based on those models is described.

Keywords
CAD/CAM models, Artificial intelligence, Intelligent knowledge-based systems, Intelligent manufacturing systems, Knowledge based systems.

1 INTRODUCTION

The known formula (Wirth, N., 1976): »Algorithms + Data Structures = Programs » shows that within the scope of software there are two sources, two antithesis's, which coexist in dialectic unity and struggle. One has a command, algorithmic type and another ones declarative, nonprocedural, based on models. At the dawn of computer science there was a complete domination of an algorithmic source. This source is alien for the style of human thinking. The engineering books have no algorithmic form. That has resulted in occurrence in a new trade - programmers. The programmers for a long time have pushed aside nonprofessionals from computers.

Only the arrival of personal computers with their simple and accessible graphic interface has allowed, at last, wide circles of nonprofessionals to sit to the

G. Jacucci et al. (eds.), *Globalization of Manufacturing in the Digital Communications Era of the 21st Century*

computer monitor. However, they are not admitted into the treasury of computer science temple - in the shop, where software is developed.

The mankind enters century of computer science. In the near future everyone should have the personal workplace equipped with not only universal software, such as the text or graphic editor, but specialised systems, which have to be filled by personal knowledge and experience. Programmers never will cope with this tremendous task. They should admit nonprofessionals into the treasury of them temple. For this purpose it will be necessary to replace algorithms by models and to create the appropriate methodology.

2 MODELS VERSUS ALGORITHMS

Models and algorithms are fundamental concepts in mathematics and computer science. A model is a set of abstract objects or some sets of abstract objects of different nature distinguished by names with a system of the relations between elements of these sets. Algorithm is an exact instruction of a sequence of actions, necessary for obtain the required result. Nowadays model, in most cases, is used only for a formal description of calculations objects, defining **WHAT** is necessary to calculate. Algorithm as the basis of computing process determines **HOW** it is necessary to calculate.

It is absolutely clear, that without stating what is necessary to determine, the process of calculation has no sense. Therefore without model is not to do. The question on a role and place of the algorithm is not so simple as it seems on the first sight and represents the greatest interest. Without the algorithm, certainly, is not to do too, however using methodology of artificial intelligence, it can be excluded from the task specification.

The history of information technology development witnesses that the models slowly, but steadily restrict algorithms.

At the first stage of this development the computers were used for the decision of difficult computing tasks. The programs were composed with the operators of control and data processing, and also some kind of embryo of model - a section of the data description containing not structured set of variables.

At the second stage computers were applied to automate the management and processing of large data volumes. There was involved a concept of databases, relied upon mathematical models of data structures. So the models formed with the help of languages of the data description, for the first time have been put into practice of programming.

The works on creation of intelligent robots have served as an accelerator for the beginning of third stage. At this stage a concept of knowledge bases relied upon on of artificial intelligence theory has been introduced. There were developed the expert systems using various models of the knowledge representation, necessary for the decision making. The models have expanded its territory, stepped out from area of passive, in area of active information resources. As to algorithms, they begun to turn into invariant software - operational environments ensuring the decision of tasks, formulated by the experts in the languages of models.

Thus, if at the first stage one can consider data definition as an embryo of model in an algorithmic body, knowledge engine at the third stage represents a remainder of algorithm in a body of models.

3 SYSTEMIC CLASSIFICATION OF MODELS

Any model includes two basic components: a set of objects and a set of relations.

$$\mathbf{M} = < \mathbf{A}; \mathbf{R}_1^{s1}, \dots, \mathbf{R}_n^{sn} > \qquad (1)$$

Here **M** is a model, **A** is a carrier of the model, $\mathbf{Ri}^{si}$ is *i*th relation (i = 1..., n) in the relations set that forms a structure of the model. If we use a type of the carrier as criterion of classification, we can obtain the traditional division of sciences, anyone of which is engaged in investigation of the certain type of elements (physics, chemistry, mechanical engineering, electronics etc.). One can develop completely other classification of systems with using the relations as criterion. The science studying systems in this aspect, is called systemology (Klir, G.J., 1985). From the systemic point of view the hierarchy of classes define epistemologic levels, i.e. levels of knowledge (table 1).

The lowerst level in this hierarchy designated as a level 0 system is recognised by the researcher as such. At this level the system is defined through the set of properties and carries the name *initial system.* Other words at a level 0 there is considered properties of researched or projected system.

On higher epistemologic levels systems differ one from another by the level of knowledge of the variables appropriate initial system. In systems more high level is used all knowledge of systems of lower levels and besides contains the additional knowledge which inaccessible to the lowest levels.

After the initial system is complemented by the data, i.e. actual variables values, there is involved a new system. This is an initial system with the data, which dispose on 1st epistemologic level. The systems of this level refer to as *data systems.*

The 2nd level represents a level of knowledge bases for the generation of variable values, determining properties of initial system. At this level the functional relations of variables are given. Variables include ones, determined by the appropriate initial system and, probably, some additional. As a main task of this level is a generation of the initial system properties, the systems of a 2nd level refer to as *generative systems.*

On 3rd epistemologic level systems determined as generative or system of a lower levels, refer to as subsystems of common system. These subsystems can be incorporated in the sense that they have some common variables. The systems of this level are named the *structured systems.*

Table 1 Levels of knowledge representation and their models

N	*Knowledge level*	*Model name*	*Model*	*Model carrier*	*Model structure*
0	Initial system	Model of the object data (concept and its intensional)	B = <A; R^k>	The dictionary of object properties (set of attributes) A = {a_1, a_2, ...,a_k}	The relation schema (contents of concept) $R^k \subset A \times A \times \ldots \times A$ $R^k = (a_{i1}, a_{i2}, \ldots, a_{ik})$ Key of the relation $K^p \subseteq R^r$, $p \leq k$
1	Data system	Set of object copies (sample of concept volume)	E = <D_1, D_2,..., D_k; r^k>	Domains of attributes (allowable sets of attributes) $D_i = dom(a_i)$	The relation with schema R^k $r^k \subset D_1 \times D_2 \times \ldots \times D_k$ Set of corteges (set of concept denotats) $r^k = (d^i_1, d^i_2, \ldots, d^i_k)$
2	Generative system	Method of object:	P = <V_1, V_2,..., V_r; F>	Object properties and method variables	Functional relation $F \subset V_1 \times V_2 \times \ldots \times V_r$
		Mathematical nongeometrical model		Numerical constants and variables, functions, mathematical and logic relations, operators and operations	Systems of the equations and inequalities
		Geometrical model		Points, lines, surfaces, bodies	Associations, crossing, subtraction, transformation, accounts of parameters

Table 1 Levels of knowledge representation and their models (continue)

N	*Knowledge level*	*Model name*	*Model*	*Model carrier*	*Model structure*
		Model of expert knowledge (production rule)		Numerical and non-numerical variables: input Vi |u (i)= 0, output Vi |u (i)= 1, where u(i) determinants an input - output	Transformation of input variables to output ones
3	Structured system:				
	Structured initial system	Multiagent system (network of the agents)	SB = <B, A; C>	Objects and their properties B = $\{b_1, b_2, \ldots, b_l\}$ A = $\{a_1, a_2, \ldots, a_n\}$	Semantic network of objects $C \subset B \times B \times A \times \ldots \times A$
	Structured data system	Database (Relational)	SD = <B, G; J>	Objects and capacity of the relations between them $B = \{b_1, b_2, \ldots, b_l\}$ $G = \{O, P, Z, N\}$	Structure of a database $J \subset B \times B \times G$
	Structured generative system	Knowledge base (method of object)	SP = <F, V; K>	Rules and variables of models $F = \{f_1, f_2, \ldots, f_l\}$ $V = \{v_1, v_2, \ldots, v_n\}$	Semantic network of rules $K \subset F \times F \times V \times \ldots \times V$
4	Metasystem	Model of metasystem (AND / OR graph)	MB = <B, M; L>	Objects and types of connections between them $B = \{b_1, b_2, \ldots, b_l\}$ M = {AND, OR}	AND / OR graph $L \subset B \times B \times M$

On 4th epistemologic levels and higher the systems consist of a set of systems determined at a level 1, 2 or 3, and some metacharacteristics (rule, relation, procedure), describing replacements in systems of a lower levels. These are levels necessary to form conceptual AND / OR graphs.

4 LADDER OF MODELS TO TOPS OF KNOWLEDGE

The models of first two levels are well investigated in the theory of relational databases (Maier, D., 1983). The initial system is intended for the description of some object or concept, equivalent to it. The carrier of model is a set of attributes of various types. Not structured set of those attributes with the description of identifier, types and names in language of business prose form the dictionary.

Structure of 0 level model is a relation, which determine the object or the contents of concept.

The data system represents some set of copies of object or samples of concept volume determined by initial system. The carrier of 1 level model is a set of domains of attributes, and structure represent the relation with the schema described at a 0 level.

The generative system is connected to a method of object determining its behaviour. Depending on a type of models it is possible to use models (mathematical nongeometrical, geometrical and expert). The carrier of nongeometrical model is composed from numerical constants and variables, functions, mathematical and logical relations, operators and operations. The structure of a such model is formed by systems of the equations and inequalities. The carrier of geometrical model is composed by null dimensional, one dimensional, two dimensional and three dimensional elements, and the structure is formed by associations, crossing, subtraction, and also geometrical transformations and operators of accounts of lengths, areas, volumes, centres of weights, moments of inertia etc.

The expert models differ from mathematical ones by their carrier, which includes beside numerical variables also nonnumerical ones. The structure of such models is formed, usually, on the bases of production rules.

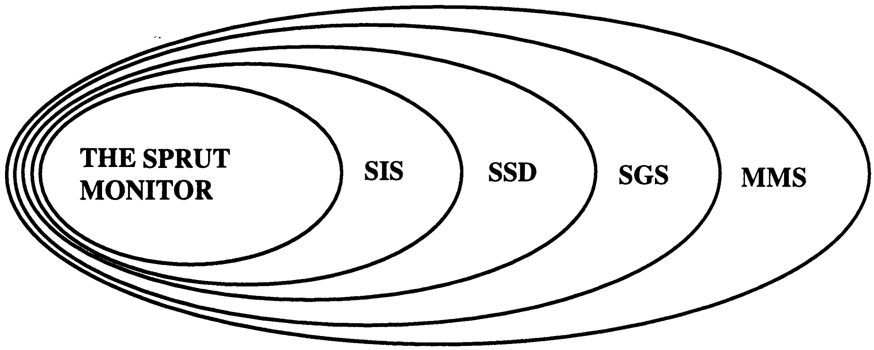


Figure 1. Hierarchy of models in integrated environment SPRUT: SIS- the structured initial system; SSD- the structured system of the data; SGS- the structured generative system; MMS- multiagent metasystem.

There are various approaches to develop of computer integrated manufacturing software in accordance with described above hierarchy of models. It is possible to connect with the help of the interface modules of database management system, packages of the programs, geometrical modeller, expert system, means of applied

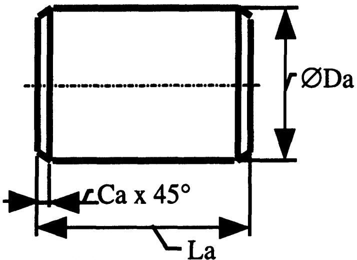
interfaces generation created by the various vendors. The long term experience show, that it is a wrong way.
In Russian SPRUT technology the creation of a hierarchy of knowledge models is conducted on a uniform environment basis (figure 1). A kernel of the system is the SPRUT monitor, which serves to represent knowledge on various level for the development of specialised subsystems and for the decision of tasks on the basis of the generated models.

Table 2 Mathematical nongeometrical models

The name of model	*Representation of model in input language UniCalc®*
System of the nonlinear equations, inequalities and logic expressions	$x^3 + 10*x = y^x - 2^k$; $k*x + 7.7*y = 2.4$; $(k-1)^{y+1} < 10$; ln(y+2*x+12) < k+5 or $y>k^2$ --> $x < 0$ and $y < 1$; $x < 0$ --> $k > 3$;

Level of generative systems is a basic arena of struggle between models and algorithms. Until recently it seemed that the applied expert does not have any other opportunity as by writing the model, necessary to him, i.e. system of the equations and inequalities, trying to search in calculus mathematics for suitable algorithm of the decision. When there is not such algorithm he have to select approximation of the model by another one, for which the algorithm exist.
The intelligent solver of mathematical tasks UniCalc allows abolish algorithms in this area (Narin'yani, A. S., 1991). Thus the form of knowledge representation practically does not differ from standard mathematical, as it can be seen from table 2.
In SPRUT technology geometrical knowledge is represented in the form of the subroutines in geometrical language. Formal variables of these subroutines correspond to the dimensions of the object drawing, which is formed as a result of work of knowledge bases, which are carrying out parametrical synthesis. In table 3 is given an example of geometrical knowledge representation.
In SPRUT technology there are some ways of creation of the geometrical subroutines. In a usual mode programs are spelled and debugged with the help of the monitor of environment. There is a visual geometrical programming mode when programs are generated automatically. Last way relieves the user of necessity to know language of geometrical modelling.

Table 3 Geometrical models

Elements	*Geometrical Knowledge*	*Geometrical Models*	
		Two dimensional	*Three dimensional*
Rotary parts	Forming contour of a part  Da = SET[1,1]; La =SET[1,2]; Ca = SET[1,3]; P1=0,0; L1=Ly,M0; L2=X(Ca),Y(Da/2),A45; L3=Lx,M(Da/2); L4=X(La-Ca),Y(Da/2),A45; L5=Ly,M(-La);L6=Lx,M0; P2=L5,L6; K1=P1,L1,L2,L3,L4,L5,P2; TOSET(2)=K1	! Surface of rotation P1=0,0; P3=X(0.001),Y0; C1=X0, Y0,R(0.001); K2=P3,C1,P3; K1=SET[2],P1,P1, A90; MATR 1=ROT,X, A90; CC 1=K1; CC 2=K2; SS 1= RADIAL, BASES=CC 2, DRIVES=CC 1, STEP 0.0002,0.5; AS i = SS 1	! Body of rotation P3=0, (SET[1,1]/2- SET[1,3]); P3D 1=0,0,0; P3D 2= SET[1,1],0,0 SOLID(j) = ROT, P3D 1, P3D 2, Set[2], P3,M(0.5)

For representation of expert engineering knowledge it is convenient to use concept « simple production system ». Production system is represented by the triple

PS = < F, P, I >. (2)

Here **F** is a working memory of system containing the current data; **P** is a knowledge base, which contains a set of production rules; **I** is a knowledge engine.
In SPRUT technology the base of knowledge is formed from modules of engineering knowledge (Evgenev, G. B., Kovalevsky V.B., 1996). Such module represents the generalised functional block, which carries out data transformation. The most successful and widely widespread representation of functional blocks is the standard IDEF0. In this standard the functional block has a design given in a figure 2.
In production systems an element of knowledge representation is a production rule. Such rules contain preconditions, determining those applicability at the certain state of database variables (if < condition >, then < action >).
Functional blocks have to be presented in the form convenient for their definition by engineer at input of knowledge in the computer. The simplest form is a table (figure 3).

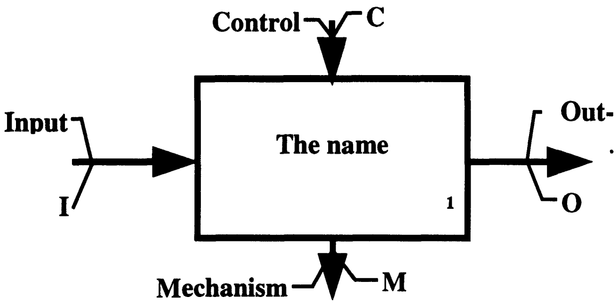


Figure 2. Functional blocks IDEF0

Such table contains all elements of the functional block submitted in a figure 2. The names of parameters should get out from the dictionary of system, as well as their names - identifiers necessary for a spelling of the formulas. The condition represents restrictions, imposed on entrance and managing parameters and functions, determining a range of definition sold by the module.

The module: <name>
The name: < the Description of function >

The name of parameter	Constraint	Identifier
1. 2.INPUT(I) AND CONTROL(C) 3.	CONDITION(C)	
4. OUTPUT(O)	MECHANISM (M)	

Figure 3. External representations of the module of engineering knowledge

The modules of engineering knowledge (MEK) should realise the following functions: setting up the values of target variables; setting up the values under the tables (figure 4); selecting the values from a database; calculation of values under the formulas; calculation of values with the help of program modules; construction of geometrical images. The mathematical and geometrical models are connected in methods of objects with the help of appropriate MEK.

The third level of knowledge includes the structured systems of the previous levels. The carrier of model of the structured data system according to the standard IDEF1 are the objects and capacities of the relations between them: **O** (zero, one or more); **P** (one or more); **Z** (zero or one); **N** (in accuracy N). The capacity show, what quantity of copies of descendant object can exist for each copy of parent object. Structure of this model defines structure of a database represented in IDEF1 as diagram of a special kind.

The model of the structured generative system has production rules and their variables as its carriers, and its structure is a semantic network, on which knowledge engine carries out procedures of a logic inference.

Module: block5
Designer: G. Evgenev
Name: assignment of a standard length
Source of the information: V. Anuriev. The handbook of the designer, v.2

The name of parameter	Constraint	Identifier
1. Part name	axis smooth, axis with collar	b1
2. Diameter of an axis standard, mm	(0 , 50]	a1
3. Length of an axis initial, mm	(0 , 300]	a12
4. Length of an axis standard, mm	TABL1	a2

TABL1

Length of an axis initial, mm	Diameter of an axis standard, mm							
	5	6	8	10	12	16,18	20	22
(25, 28]	28	28	28	28	28	28		
(28, 30]	30	30	30	30	30	30	30	

Figure 4. External representations of the module - table

5 AGENT AS A CORNER STONE OF NEW PARADIGM

The agent is a development of a well known concept of an object. The agent represents an abstraction of set of copies of subjects of the real world, having the same properties and rules of behaviour. The properties of object are described by initial system, and rule of behaviour by generative system more often structured. The state of object is defined by the list of values of its properties. The object with the meanings of all its properties determines a copy, which model is a cortege of the appropriate relation from the data system. Object properties include its identifier, and also indicating, describing and auxiliary attributes. Last two types of attributes are divided in relation to a method of object on input and output (figure 5). The descriptive attributes define properties internally inherent to object, and auxiliary — its structural connections with copies of others object.

The metasystem contains knowledge of all set of its versions of the decisions, known by the authors. The designing is started by input of the initial data in the agent of a highest level (for example, A on figure 5). Then operational environment converts metasystem knowlelge into relational and graphic data containing the project of one copy of a product, meeting the requirements of the initial data. This conversion is a result of process of each agent method work and transfer data from one agent to another.

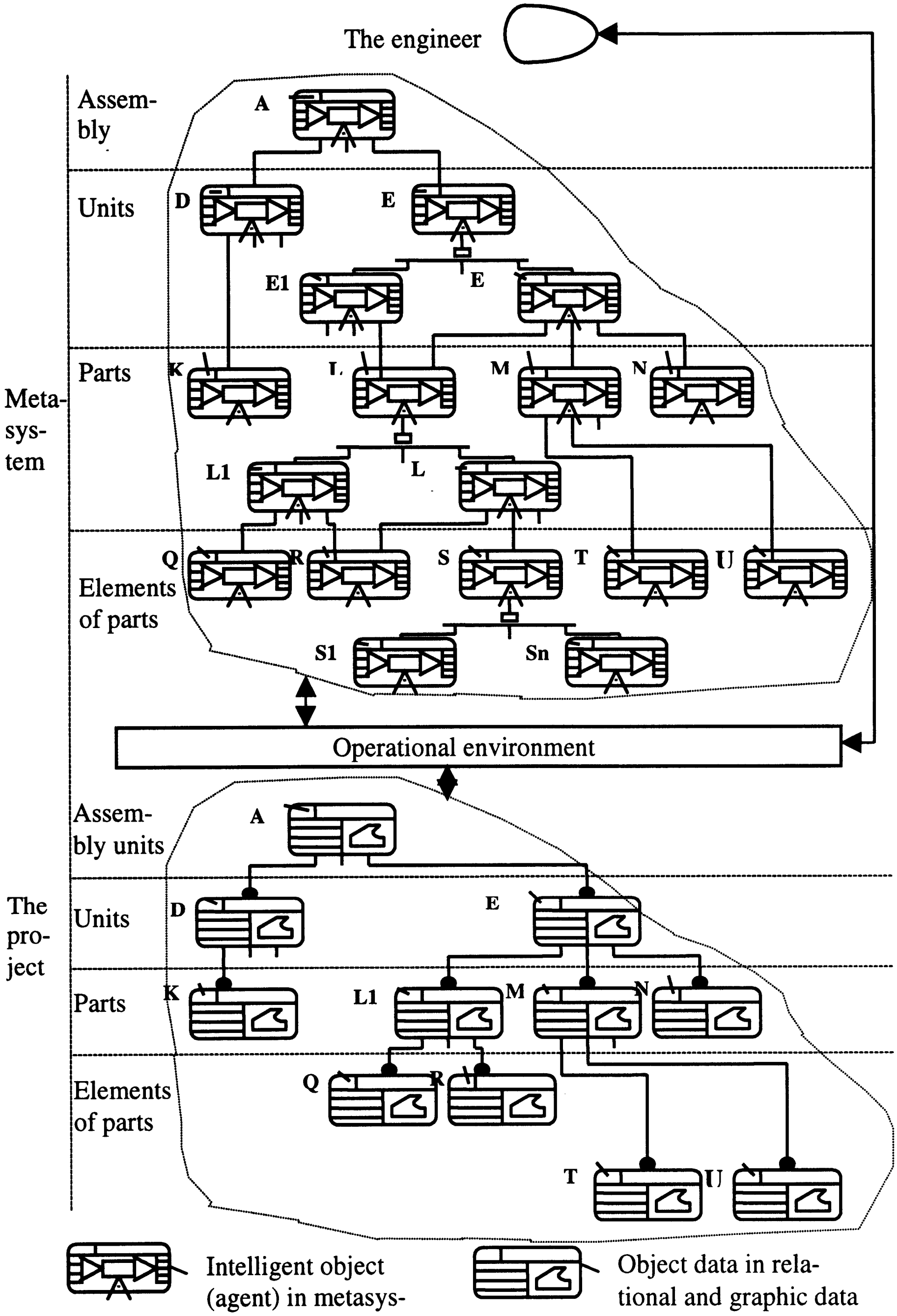


Figure 5. Multiagent metasystem

6 REFERENCES

Evgenev G.B., Kovalevsky V.B. (1996). SPRUT - Integrated Environment for Engineering Knowledge Computer Processing. *Proceedings of International Conference "Information Technology in Design" - EWITD'96*, p.p. 38-45. ICSTI. Moscow.

Klir, G.J. (1985) Architecture of systems problem solving. Plenum Press, New York, London.

Maier, D., (1983) The Theory of Relational Data bases. Computer Science Press, Inc., New York.

Narin'yani, A. S. (1991) Intelligent software technology for the new decade // *Communications of the ACM* – Vol. 34, No. 6. – p.p. 60-67.

Wirth, N. (1976) Algorithms + Data Structures = Programs. Prentice - Hall, Inc., Englewood Cliffs, N.J.

7 BIOGRAPHY

Georgy B. Evgenev was born 9 July 1938 in Moscow, Russia. Mechanical engineer, BS, MS degrees have been received at Moscow Aircraft Technology Institute in 1960. From 1960 to 1991 he has been employee at the enterprises of air-cosmic industry. Now he is professor of Computer - Aided Systems Department at Bauman Moscow State Technical University. Doctor of science (Eng.) grade he received in 1978. From 1993 he is Academician of International Informatization Academy. He has experience in NC part - programming, CAE, CAD - CAM, CAPP, CIM systems, Artificial Intelligence, software tools and environments.