HYPERINTELLIGENCE FOR SYNTHESIS OF STRUCTURED IMAGES OF TECHNICAL OBJECTS

V. P. Zhidakov

The paper presents the results of some studies of logical-structural modeling methods which have led to formalization of parallel-sequential algorithms for synthesis of structured images of technical systems. The studies have produced a mathematical model of hyperintelligence for interaction with a team of developers.

INTRODUCTION

The paper presents algorithmic foundations and applied aspects of hyperintelligence for solving complex problems — synthesis of images of technical products. The overall aim of hyperintelligence ("global intelligence") is to create a collective "brain" by using distributed data processing tools linked by digital telecommunication and an appropriate complex of programs that support goal-directed interaction of information and computing resources, and also using the accumulated knowledge in a given application domain with large social and scientific organizations.

This problem is particularly relevant because of the wide use of local-area and large-scale computer networks and the need to increase the efficiency of computer applications for solving complex and very complex problems associated with the development of complex technical systems. This direction of research and development relies on the notion of design as a process of accumulation of interconnected information about the interaction of developers with an information/computing system, when computational problems are solved as a means for refining the completeness and quality of the accumulated information [1], and also on the concept of collective computations [2] imposed on the functioning of the design organization whose entire activity is focused on the development of highly complex engineering objects. The algebra of semigroups, algorithmic algebras [3], and the theory of correspondences as presented in [4] have all been used as formalization languages for this purpose.

FOUNDATIONS OF LOGICAL-STRUCTURAL MODELING

A machine image of a complex object is mapped to memory, which is a set of storage locations O for discrete units of information. One component of the image is defined on the set of functions \( f = \{ f_1, f_2, ..., f_n \} \) and exists in the form of a structure or a subset of the direct product \( O \times f \). In the design process, the elements \( f_i \in f \) should be interlinked by some set of linking functions \( f^\prime = \{ f_1^\prime, f_2^\prime, ..., f_m^\prime \} \); a structure is defined on this set as a subset of the direct product \( O \times f^\prime \). Application of control functions produces the relation \( G_s^k \subset f \times f^\prime \), and the structure is \( S = O \times G_s^k \).

In order to construct the second component, we introduce the set of elements \( \Phi = \{ \Phi_1, \Phi_2, ... \} \) which realize functions in the form of physical processes and a set of physical linking elements \( \Phi^\prime = \{ \Phi_1^\prime, \Phi_2^\prime, ... \} \) such that a structure \( G_s^\prime \subset \Phi \times \Phi^\prime \) is obtained, which exists in the form \( S^\prime = O \times G_s^\prime \). Similar sets, relations, and structures are formed for other components of images using the elements \( K = \{ K_1, K_2, ... \} \) and \( K^\prime = \{ K_1^\prime, K_2^\prime, ... \} \) from which the finished product is made. Here \( K \) are the constructive elements of the product and \( K^\prime \) are the linking constructive elements. We have the linking structure \( G_s^\prime \subset K \times K^\prime \), and the overall constructive structure exists in the form \( S^* = O \times G_s^* \).

The total accumulated information structure in the storage medium is \( S_{TP} = (S \cup S^\prime \cup S^* \cup ...) \times S^\prime \), where \( S^\prime = \{ S_1^\prime, S_2^\prime, ... \} \) is the set of interstructure linking elements. The structure is specified in the form of a programmable table \( O_T \), representing the sets A, B and the environment O where the relation graphs on these sets are formed. Programming of the table \( O_T \) is a process which determines the sets A, B, O, produces the graphs \{ \( G_1, G_2, G_3, ... \) \} that specify the structures and the derived struc-
tures, and defines the procedure $G_0$ that creates a structure when the sets $A$, $B$, $G_1$, $G_2$, $G_3$,... are superimposed on the set $O_1$. 

Single or multiple application of the procedure $G^O$ to the pair of sets $A$ and $B$ generates the graphs $G^O \subseteq A \times B$, $G_1^O \subseteq A \times A$, and $G_2^O \subseteq B \times B$.

The contents $M$ of a programmed table is the result of its programming, written as $M \otimes G_k^O(A, B)$. Here $\otimes$ stands for application of two operations: direct product and subset extraction; $k$ is the number of times the procedure $G^O$ is applied. The total number of steps to construct a graph from a pair of sets is given by the asymptotic bound $\lambda^O = 2^{mn}(m = \text{Card } A, n = \text{Card } B)$. The particular case of table programming generates correspondences, i.e., $M \otimes G_1^O(A, B) \equiv (G, A, B)$. In this case $\Gamma = (G, A, B)$ has a whole range of useful properties. For instance, if $A$ and $B$ are sets which are geometrically represented by discrete coordinates of two parallel planes in space, then $\Gamma$ can be directly transformed into an image of a spatial geometrical figure. Application of the composition operation to several correspondences makes it possible to represent in coded form the image of complex geometrical figures.

Table programming is a convergent iterative parallel-sequential process [5]. It realizes a design algorithm or model by solving a system of interconnected problems: constructing circuit diagrams (as a form of structure); creating a structure that is typical of the system (the problem of engineering design of the image), etc. When designing a complex product, similar models are simulated for the design of the entire unit, parts of the unit, technical equipment, complex fitting, etc. The number of times the program is replicated in the parallel-sequential development process is determined by an integral bound, which for realistic processes may reach hundreds of thousands.

**MICROSTRUCTURES AND MACROSTRUCTURES OF LOGICAL-STRUCTURAL MODELS**

Logical-structural models for realization of information/computation processes considered on the logical level are a system of special automata connected in a certain order and formally representable by some microstructures [6]. Combinations of simple microstructures are tuned by creating graphs that model Boolean functions, predicate functions, integer functions, and tables. For instance, if $X = \{0, 1\}$, $Y = \{0, 1\}$, then $G \subset X \times Y$ subject to functionality conditions is a graph of a Boolean function of a single variable. $pr_1G$ and $pr_2G$ can be modeled by the states of the input and output register, and the graph itself can be modeled by a storage register array.

More complex functions are realized by complex macrostructures. The synthesis of macrostructures of logical-structural models is based on the properties of the composition of correspondences and the table linking procedure. Programs using the operations of composition, decomposition, folding in a ring, absorption, shift, etc., are executed on macrostructures. Graph programming and certain connection of microstructures can be used to realize various devices, such as a generator of the coordinates of points in space or a converter of structural solutions to a graphic image on point sets.

Macrostructures are formed from microstructures by some standard techniques. For example, creating three tables $\Gamma_1 = (G_1, A, A)$, $\Gamma_2 = (G_2, A, X)$, $\Gamma_3 = (G_3, A, Y)$ on the set $A$ and programming them in microstructures, we can construct a generator of point coordinates by a simple cascade connection of microstructures. A special class of logical-structural models are the information macrostructures. They are used for accumulating complex structural solutions and for performing a number of invariant transformations on such structural solutions. For example, a simple cascade connection of identical microstructures models composition, which can be used to realize the operations of folding in a ring, cutting a ring, absorption and extension of the information space.

A complete logical-structural model is a schema and a tuning program, which are constructed by hierarchical levels: microstructure, macrostructure, a system of macrostructures. Different functional blocks are formed on these levels: for accumulation, storage, and updating of structural solutions; for processing (transformation and deformation) of structural solutions; for control of the process of logical-structural modeling on different levels.

A general mathematical model of actions on an array of information storage locations is provided by the system of predicates. Such predicates are defined on the set $\Omega$, designated for the design of the structure needed for microstructure tuning, and the subset $A \subset \Omega$ which stores the structure formed on the information set $\{I\}_p = \{I_1, I_2, ..., I_j, ..., I_k\}$.

The following actions are performed when constructing the image of the table $\Gamma_\Omega$ on the set $O$: the sets $H^\Omega$ and $H^O$ are determined — the value sets of the predicates associated to the elements of the storage medium and used for pointing to the elements of the sets $\Omega$ and $O$; the tables $\Gamma_1 = (G_1, H^\Omega, O)$, $\Gamma_2 = (G_2, H^O, O)$ are created; predicates are generated which, given some initial data, point to the corresponding locations in the media $O$ and $\Omega$; value sets $H^\Omega_{\text{init}}$ and $H^O_{\text{init}}$ of the predicates are formed, whose elements point to information flows when selecting from medium $\Omega$ for transfer to medium $O$; the tables $\Gamma_{\text{IH}} = (G_{\text{IH}}, H^\Omega_{\text{init}}, I)$ and $\Gamma'_{\text{IH}} = (G'_{\text{IH}}, H^O_{\text{init}}, I)$ or predicate functions based on these sets of tables are created, which given some