QUESTIONS THAT ARE RELATED TO IMITATIVE MODELING OF SOFTWARE OF INTERACTIVE SYSTEMS ON STAGES OF THEIR STRUCTURAL AND FUNCTIONAL SPECIFICATION ARE CONSIDERED. A NET OF A NEW TYPE IS INTRODUCED THAT IS INTENDED FOR MODELING AND Allows INCREASED CAPACITY OF DESCRIPTION OF INFORMATION SYSTEMS. SOME EXAMPLES ARE GIVEN.

We consider questions that are related to imitative modeling of a software of interactive systems on stages of their structural and functional specification.

In elaboration and study of complex computing systems different modeling methods are applied, which give a possibility of displaying "narrow" spots in the future system. The most effective tools for modeling complex systems are Petri nets [5]. With the help of them it is possible to reflect adequately not only the structure of the system but also the dynamics of its work.

Using Petri nets for a description of interactive dialogue systems, the following was assumed in many papers: a representation of operational structure of dialogue by Petri nets and description of composition operations over elementary nets in [1]; a description of a model of dialogue of a man with a computer by Petri nets in [2]; hierarchical nets in [4].

In the paper we give a modification of the hierarchical Petri net for modeling of a numerical process with interactive interactions, and also estimates of resources of the basic memory and availability time to different resources of the computing system.

On the basis of the net that was introduced, we construct a formal model and consider optimal algorithms of control of a computing process in the system.

Modification of the hierarchical net. Let $D$ be a fixed alphabet, and $D^*$ a set of words that are generated from letters of the alphabet $D$. We call a set $A \subseteq D^*$ the controlling dictionary.

The set of arcs in the Petri net is divided into two subsets: $E_1 = \{P \rightarrow T\}$, $E_2 = \{T \rightarrow P\}$. We make a special marking of elements of $E = E_1 \cup E_2$ of the net. We assign to each element of $E$ a memory $M$. One word as well as several words can be in the memory.

Let functions $f: T_1 \rightarrow D$, $h: T_2 \rightarrow \Sigma$ be labeled functions over the alphabets $D$ and $\Sigma$, respectively. Sequences of operations of simple transitions generate words from the set $A$. A function $d: \{P \times T\} \rightarrow A$ writes words from $A$ into the memory of elements of $\{P \times T\}$.

The rule of generation of words: if a composite transition is passive, then on the output of the transition on each arc from the set $\{T \times P\}$ there exists the empty set $\lambda$; each operation of an elementary inner transition generates a symbol from the alphabet $D$, by which the transition is marked. The symbol is written in the memory of the output arc. As the result of the sequence of operations a word from $A$ appears on the output arc.

On the appearance of marker in a position and startup of transition additional assumptions are imposed:

A1. Appearance of marker in a position: it appears if the word on the input arc is equal to the word in the memory of the output arc.

A2. Startup of transition. The transition can be transformed into an active state, i.e., the marker can start up the transition if words on arcs of the input position are equal. If the word in the memory of the arc is empty, then any realization can function.
We define a net in the following way: $N = (P, T, E, \mu, \{M_i\}, A, d, f, h)$, where $P, T, E, \mu$ are elements of the traditional net, and the elements $\{M_i\}, A, d, f, h$ are introduced above.

**Model of an interactive system.** Suppose that $J_i$ are states of the system, which we shall represent by elements of the set $P$ of the Petri net. For elements of the set $T$ we assign in correspondence the task for the system under the transition from the state $I_i$ into $I_{i+1}$.

We assume that for some tasks there are several alternative ways of realization. As a task we understand, for instance, loading, realization and release of memory by a problem module. A complex transition models the task of the computing module. For instance, activation of a complex transition is a load of the module into the operating memory, the task of any internal net of a complex transition is a modeling of computations of the loaded module, completion of the task of a composite transition is a release of the resource of the basic memory.

Each realization of a composite transition is characterized by the length of the word $L_j$ generated by it, and also by the number of interactions $\Psi_j$. The entire algorithm of work of the system will be estimated by the values $\Psi_0$ and $L_0$.

Very often there appear problems in which we need to organize the computing process in such a way that the number of interactions is defined in it. If we model the problem by a net, then the problem is reduced to a definition of trajectory of operating realizations of composite transitions, where a definite number of arcs should be included in the trajectory, which are marked by controlling words.

We shall look for a solution to the problem in the form of a vector $a$ of dimension $N$, where $a(i) = j$ means that for the $i$th module the $j$th realization is chosen.

In order to solve the problem stated, we use the algorithm of implicit successive tests. As is known, this algorithm relies on the generation of a sequence of solutions and on probing them. A particular solution is called a vector for which some values are fixed in the process of solution. The remaining coordinates of the solution vectors are free. A supplement of the partial vector given will be a solution that is defined by a partial solution together with assignment of values to free variables.

Each partial solution defines a value of the function $\Psi_k$ that is equal to the number of interactions in the solution vector, and $L_k$ is equal to the length of word that is generated by the vector solution which is given. The function has the form $\Psi = \sum_{i=1}^{N} a_i \in A^a(i)$, where $l$ is the variable that defines interactions.

In checking for optimality for the vector of partial solutions the values $\Psi_k, L_k$ can be evaluated in the following way: for the fixed coordinates of the vector we take their values $L_j$, for the free coordinates we take minimal values of lengths of words for all realizations of composite transitions.

In the case where one of the conditions $L_k \leq L_0, d_1 \leq \Psi_k \leq d_2, d_1, d_2 \in N$ is not satisfied, the partial solution is excluded from further considerations.

The algorithm defining the trajectory with optimal number of interactions is realized in the language of the programming system Fortran EC in the medium OC6.1. With its help in the package of applied programs for the system of computation and manufacture of sheet metal of aircrafts the optimal trajectories of computing process are defined with an interval given of interactive interactions of a user and a computer.

**Algorithm for defining optimal trajectories of computing processes in a system.** We define the optimal sequence of operations of transitions in the net. Each transition generates a word $L_t$ during the work. We calculate the length of the word $\Phi(L_t) = l_t$. The length of the word characterizes the volume taken by the memory module.

Let a net with $N$ transitions and $M$ positions be given, and let an initial marking $\mu_0$ of the net be given too. The marking $\mu_k$ is finite.

Let $D$ be an incidence matrix of transitions and positions in the net. The dimension of the matrix is $(M \times N)$

$$d_{ij} = \begin{cases} 1, & p_i \in \text{in}(t_j) \\ -1, & p_i \in \text{out}(t_j) \\ 0, & p_i \notin \text{in}(t_j) \& p_i \notin \text{out}(t_j) \end{cases}$$

where $\text{in}(t_j)$ are input positions of the transition $t_j$, $\text{out}(t_j)$ are output positions of the transition $t_j$. We write the matrix equation for the net: $\mu_k = \mu_0 + D \cdot \Psi_k$.

A solution to the equation is the set of Parih vectors $\{V_p\}$. In order to define the optimal trajectory we come to a