Families of Alternative Stochastic Action Networks: Use for Process Science

Alexander Geyda
St.-Petersburg Federal Research Center of the Russian Academy of Sciences
St. Petersburg, Russia
geida@iias.spb.su

***Abstract*—We consider a new mathematical formalism, which allows us to model alternative functioning cases due to changing system environment conditions and the environment's impacts. Models: to research dynamic capability organizational capability, sustainable development, information technology capability of the system, can be created based on suggested families of alternative stochastic action networks. Such families are complex of basic graph-theoretic objects and mappings between them. They describe functioning in terms of the alternative action networks, which can be interrupted and alternated. Such alternations form families of alternative stochastic action networks. We plan to built and use suggested models using process mining and process science techniques.**

I. Introduction

There is a gap between the need to decide a variety of problems, such as infonomics (1, 2), information technology (IT) governance (3), process mining (4) problems, on the one hand, and theoretical models and methods available for such problems decision as related mathematical problems - on the other hand. We suggest system capability analytical research methods and models (5), to fill this gap. Among such models and methods, families of alternative stochastic action networks are suggested. System capability is a system's ability to achieve changing goals as a reaction to changing environments (6). We consider complex technical systems (CTS), which are such systems that include interrelated elements of a different nature, i.e., mechanical, organizational, human, and technological components. For such CTS system capability is required to react correctly on CTS environment changes and impacts (7–9), particularly - to respond to environment attacks, to respond on goals changes so - to interact properly with the environment and with parts of the system which is under environment impacts. Information operations are needed (10) to create such capability and to provide interaction under changed conditions. The system capability is used to estimate information technologies performance indicators, dynamic capabilities indicators, organizational capabilities indicators, system dependability indicators. Further, indicators of this property are used to solve various practical problems as appropriate mathematical problems of indicators estimation and the CTS elements, capabilities, information operations synthesis based on indicators, estimated as a function of possible CTS characteristics. For estimation of such property complex of models is necessary (11). It shall reflect the interacting system, its environments of a different kind, and information operations. Information operations are required to check the system and its environment functioning states to measure their correspondence. Then information operations are used to alternate the CTS functioning to achieve a possibly changed goal set by changing the environment and with changing impact on the CTS. Models of information operations use as a reaction on modeled environment changes were considered in previous publications (12, 13). This publication concentrates on families of alternative action networks, which model the system's possible responses on environment changes and corresponding information operations results.

Such alternated activities models as a result of environment impact are required (6, 14–21), but not yet described in sufficient details (10, 22, 23).

II. Alternated System Functioning Graph-Theoretic Modeling

We suggest $FASAN-$ families of alternative stochastic action networks allow alternate system functioning modeling. $FASAN$ is a system of graphs, kind of Echgraphs (24). It was created based on sets of graphs (the base of $FASAN$), relations between them, and mappings between graphs. Such relations and mappings are defined in such a way families (related sequences) of alternated action networks, and mappings between networks and states (cuts of networks) are defined. Then, mapping states of the system and its environment to the new network in the family of networks is defined. As a result, trees of possible action networks and their alternations formed.

Precisely, in $FASAN$ The base is Hypergraphs edges (sets), which describe states and relations between them, and Stochastic Action Networks, which represent sets of actions and relations between them case actions are not alternated.

With the use of the $FASAN$ it is possible to describe alternations of the functioning (25), including alternations, which are defined by networks of operations. The network's alternation case is mapping the network actions to the complex state (modeled as hypergraph of states). For such mappings, we developed algorithms to map alternative networks to alternative states. Next, such algorithms allow forming trees of complex States - depending on possible alternative scenarios. We use networks of operations with start and finish vertices. The network of operations is the Directed Acyclic Graph (DAG), such that each vertex is associated with an action (operation) on individual workplaces of the system. Start vertex associated with the operation of waiting to start at the required moment. It has no incoming edges. Finish operation is the operation of waiting to report the results of actions. We plan to use the theoretical formalism of $FASAN$ with process mining techniques (4). Example of the network of operations and the maximal cuts of this network shown in Fig. 1. There are eight operations. Maximum cuts of the given network are schematically shown in Fig. 1 with dotted lines. All possible cuts are represented as all maximal cuts and their subsets. The possible cuts allow us to specify possible states of the system functioning alternations. Each dotted line corresponds

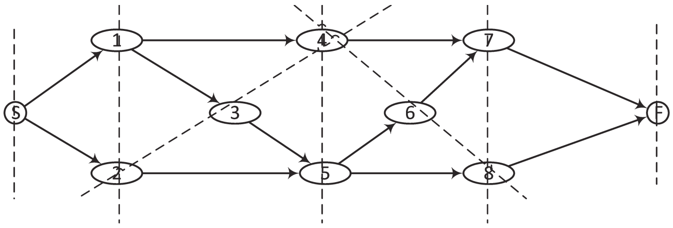


Fig. 1. Maximal Cuts of the Network of Operation

to a maximal cut. Its subsets may be cuts as well - due to possible waiting operations. It is possible to include waiting in the network explicitly. In this way, potential waiting included into any incoming edge if there are more than one incoming edges, which lead to the same ending operation - as in Fig. 2 Such waiting operations appear as a result of modeling

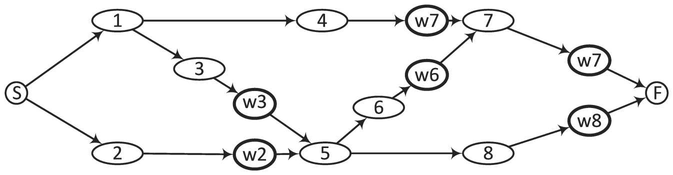


Fig. 2. The Network of Operations with Added Waiting Operations

assumptions: once an operation may start, it will start without any waiting. Thus, if only one edge is in the vertex's input set in the initial network, this edge can not split with waiting and vice versa. Alternatively, each subset of any maximal cut is considered, but some subsets realization's probabilities can be 0 by default. These probabilities are counted based on the same assumption, but in functional form: "once operation may start, it will start without waiting time equal to 0". So, the probability of such waiting to be realized will always be 0. Under other assumptions, such waiting and the corresponding subset of maximum cut may be possible, The network of technological information and non-information operations with waits shown in Fig. 3. Based on this model, the system complex states and transitions graph-theoretic model build. It

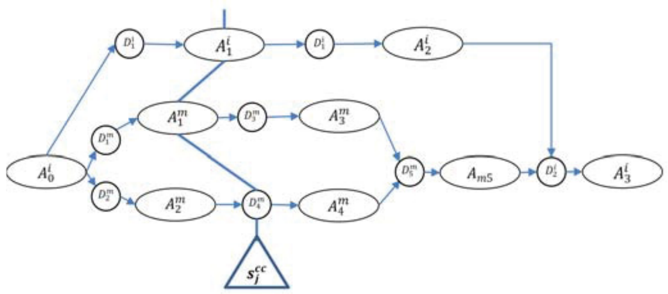


Fig. 3. The example of network with delays and possible cutting

allows to specify: $\mathbf{S}_i^s-$ the set of the possible system states under condition that the vector S_i^e of states, required by the system environment at T_i fixed; Each system state is associated with b_{is}^s- $is-$th branch at the tree T_i of possible branches of the simultaneously performed technological operations. It is created for the fixed S_i^e. Each branch b_{is}^s associated with the subset of A_u^* of information and non-information technological operations and waits for operations, $b_{is}^s \sim A_u, u = \overline{1,U}$, each one performed (or waits) at the workplace w_u. This allows to compute the possible states of the system, which corresponds to b_{is}^s. The tree T_i fragment is shown in Fig. 4. The fragment built under the condition, that technological

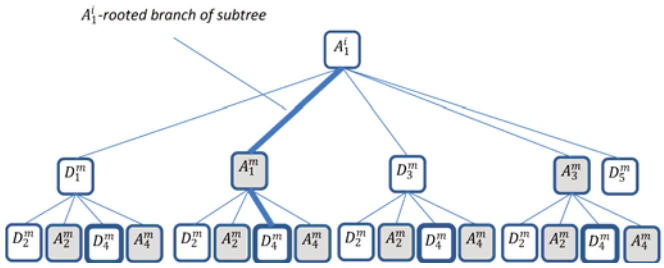


Fig. 4. The fragment of the tree of initial technological operation network cuts

information operation A_i^i is performed. Other fragments have the same structure and correspond to the cases when the other technological information operations performed. As a result, the complex tree built based on its fragments. It shown at Fig. 5. One of the possible cuttings shown. It corresponds to one of the possible alternations resulting from the network of actions and the complex system state mapping m_j^{ns} from network to complex state s_j^{se} of the system its environment, obtained with information operations use. This (first) information operation result is the state of the system and its environment descriptions. With these state characteristics, the measures of their correspondence to the demands of the environment are computed. Computations are defined by functional models, which are described further. Next,the mapping $m_u^s n$ from the system and its environment complex state s_j^{se} to new (alternated) network $N_u(s_j^{se})$ realized. This mapping is provided with information operation too. This (second) information operation results are prescriptions and descriptions needed to fulfill a new, alternated network. Such prescriptions and descriptions form the state of the new (alternated) network start. Prescriptions and descriptions quality determined by future correspondence measures at further moments of the system functioning. Such measures, under The mappings repeated, taking into account the environmental impact and its changes. As a result, the branches of $FASAN$ trees formed. Part of such branch with the two alternations, two states, four mappings, and two alternate networks shown in Fig. 6.

The trees built allows specifying and further, to computing the possible states of the system during its functioning to reach the requirements of the environment. These states used to calculate probabilistic measures of system effects and environment demands compliance.

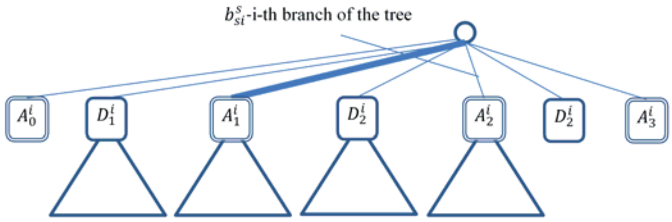


Fig. 5. The complex tree with all fragments included

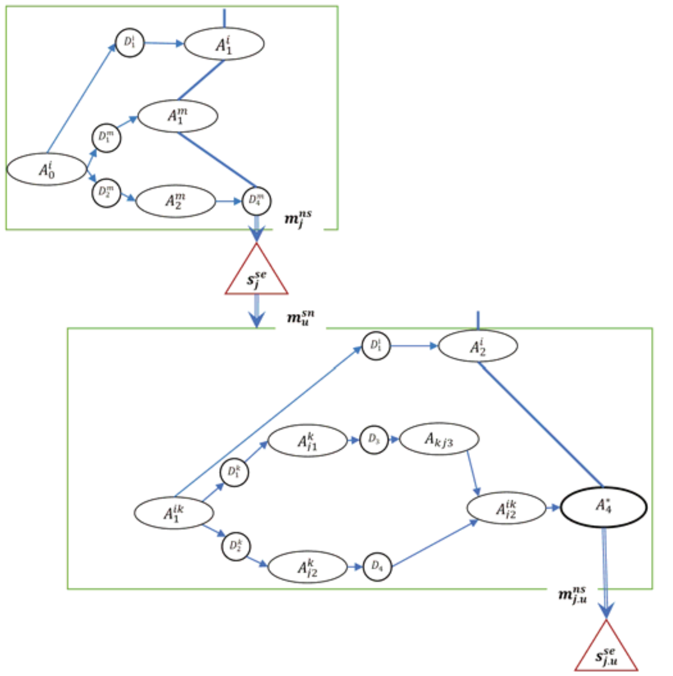


Fig. 6. The FASAN branch fragment with two networks and two complex states

III. PROCEDURE TO COMPUTE THE PROBABILITIES OF OPERATIONS REALIZATIONS

Each operation mode a_i realization and thus, state s_i of corresponding workplace at given moment characterized by probability of its actuality $p_i^a := p_i^b p_i^{ne}$ at moment T_m, where p_i^a- probability of complex event $\hat{A}_i$: the operation (action) at workplace wp_n used to fulfill a_i actual (performing), thus - operation began but not ended. Moment T_m is the moment of technological information operation fulfillment;

$\hat{A}_i = (\hat{B}_{vi} \cap \hat{N}_i / \hat{B}_{vi})$, where $\hat{N}_i-$ the operation not ended, the complex event a_i not ended under the condition all previous operations including a_i began;

$\hat{B}_{vi}$ is a complex event, which is the operation began. It consists of a_i beginning under the condition all previous operations from the set V_i of operations, previous operations to a_i-th operation, began.

Let us designate $p_i^a = P(\hat{A}_i)$, $p_i^b = P(\hat{B}_i)$, $p_i^{ne} = P(\hat{N}_i)$, $F_i^b(T_m)$- cumulative probability function value of the distribution of the moment of a_i beginning. Value counted for the moment T_m of technological information operation. This moment assumed to be deterministic. $F_i^e(T_m)$- cumulative probability function value of the distribution of the moment of a_i ending. Value counted for the moment T_m of technological information operation. States actualization probabilities depend on previous operations probabilities in a variety of ways, considered below. We consider dependencies in chains of operations and dependencies in network cuts caused by appropriate chains.

For network operations dependencies, we consider sequence (chain), wait and corresponding join, fork case, start and finish network substructure cases and their possible combinations, which are:

1. StartFork
2. StartSeq
3. SeqSeq
4. SeqFork
5. Wait
6. JoinSeq
7. JoinFork
8. SeqFin
9. JoinFin

For all vertices v_i, actualization probability p_i^a of activity, associated with this vertex is:

$p_i^a = p_i^b p_i^{ne}$, where $p_i^{ne} = 1 - p_i^e$.

p_i^b, p_i^{ne} and further, cuts probabilities, evaluated depending vertex type.

1. For StartFork structures, associated with $s = v_i, i = 0$

$p_i^b = 1$; $p_i^e = 1$,

and moments of next actions start, associated with vertices $v_k$in the list N_i of next vertices of the Fork set to 0.

Cumulative conditional (c) probabilities double list created $\langle (p_k^c, k) \rangle$ with one pair $(p_k^c, 0)$.

$p_k^c = F_i^b(T_m) = 1$ and double list copied in the same way for all elements of next vertices set N_i.

2. For StartSeq structures, associated with $s = v_i, i = 0$

$p_i^b = 1$; $p_i^e = 1$,

and moment of next vertex v_k start from the set of next vertices set to 0.

Cumulative conditional probability $\langle (p_n^c, n) n \in \overline{1, N} \rangle$ double list of the next vertex action is set to the $\langle (p_0^c = F_i^b(T_m) = 1, 0) \rangle$, and the moment of the next vertex start set to 0.

(1) and (2) cases differ when probabilities of cuts of the network are evaluated.

3. For SeqSeq structures associated with $v_i, i > 0$,

$p_i^b(T_m) = \prod_{j \in C_i} F_j^b(T_m)$, where $j \in C_i$ - numbers of operations which can began at T_m in the chain C_i of v_i, join type vertex v_j^j structure met on the way from v_i to v_j^j and all operations between v_i till v_j^j.

That means same as,

$p_i^b(T_m) = p_i^c(T_m)F_i^b(T_m)$,

Cumulative conditional (c) probability double list updated for last element $\langle(p_n^c(T_m), 0), (.), (.), (.), (p_n^c(T_m), n = N)\rangle$ in the list for the action, associated with next vertex in SeqSeq:

$p_n^c(T_m) = p_i^c(T_m)F_i^b(T_m)$.

$p_i^e(T_m) = F_i^e(T_m)$, $F_i^e(T_m)-$ normal cumulative probability distribution function of the moment $\hat{T}_i^e$ of v_i end at the given moment T_m.

The moment of action beginning, which associated with vertex v_k such that it follows (next to) v_i set to $\hat{T}_i^e$.

SeqSeq behaves differently from SeqFork when a network cuts probabilities evaluated.

4. For SeqFork vertices, all formulas are the same as for (3), but next set N_i of next vertices v_k contains more than one v_k. For each v_k same operations apply when SeqFork evaluated. Still, the double list of $\langle(p_n^c, n), n \in \overline{0, N}\rangle$ updated differently:

New pair $\langle(p_{n+1}^c, (n+1)), N+1\rangle$ added to tail of the list and updated $\langle(p_{n+1}^c, (n+1)), n \in \overline{0, N+1}\rangle$ list added for each v_k.

Last pair $(p_{n+1}^c, n+1)$ updated according formulas below:

$p_{n+1}^c = p_i^c(T_m)F_i^b(T_m)$ and copied to the all $v_k \in N_i$, where $p_i^c(T_m)$ shall be equal to p_n^c in previous to the last pair (p_n^c, n) of $\langle(p_{n+1}^c, (n+1)), n \in \overline{0, N+1}\rangle$.

Cuts probabilities are evaluated differently from SeqSeq and given below when cuts for SeqFork are considered.

5. For waits (waits always precede joins and only to joins) w_i,

Waits counted based on the same formulas of SeqSeq for double list $\langle(p_n^c, n), n \in \overline{0, N}\rangle$.

$p_i^{bw}(T_m) = \prod_{j \in C_i} F_j^b(T_m)$,i.e. probability of the beginning for them counted in the same way as for operations in chains SeqSeq, as probability of the complex event $\hat{B}_i^w/\hat{B}_{vi}$ the wait w_i began under condition all previous operation from the set V_i of operations, previous to the wait w_i , began too;

$\langle(p_n^c, n), n \in \overline{0, N}\rangle$ updated for waits like for SeqSeq.

$p_i^{ew} = \prod_{j \in V_i} F_j^{buw}(T_m)$,

$p_i^{new}(T_m) = 1 - p_i^{ew}$,

where V_i - set of other waits (excluding i-th wait), which precedes to the same join vertex, such that it is exactly the join which follows $i-$th wait (set of previous vertices of $i-$ th join except $i-$th one); $F_j^{buw}(T_m)-$ unconditional probability distribution function value of the event $\hat{B}_j^{uw}$ wait (from the set V_i) have began at T_m.

6. For JoinSeq type structures operations (which always follows waits and only waits) a_i^j, probability counted based on $\langle(p_n^c, n), n \in \overline{0, N}\rangle$ operations.

Each double list $\langle(p_n^c, n), n \in \overline{0, N}\rangle$ for each joined waits merged by pairs with same n into one new double list $\langle(p_n^{c*}, n), n \in \overline{0, N}\rangle$.

Than, $p_i^{bj} = \prod_{j \in Pr_i^w} F_j^b(T_m)$, where Pr_i^w is the set of all distinct paths of vertices to v_i with unique, not equal with other vertices on paths in Pr_i^w.

Such distinct paths p_i^{bj} corresponds to multiplication of all F_n^b or to updated $\langle(p_n^c, n), n \in \overline{0, N}\rangle$.

Each v_j is considered as one of the vertices in Pr_i^w as a unique vertex in one of the paths. Paths to v_i include paths which consist of one wait w_j^j too; All such distinct unique vertices conditional cumulative probability p_i^{bj} of all appropriate actions beginning are evaluated due to individual $F_j^b(T_m)$ multiplication.

For JoinSeq, probability of ending and not ending operation a_i^j which started as join counted just like for SeqSeq structure operations for moments T_m,

$p_i^e = F_i^e(T_m)$;

$p_i^{ne} = 1 - p_i^e$;

As usual, $p_i^a = p_i^b p_i^{ne}$ for these operations too.

7. For JoinFork structures probability p_i^{bjf} of JoinFork operation a_i^{jf} beginning, the expression takes the same form as for JoinSeq, discussed before.

For JoinForks operations a_i^{jf} end, the expression for p_i^{ejf} takes the same form as for JoinSeq.

The difference is in expressions and procedures for next operations (in the set of next for fork part), which takes the same form as for (4) and (1) and as well, differs for cuts probabilities characteristics computation.

8. For SeqFin structure operation: $p^{fb} = F_j^e(T_m)p_j^b(T_m)$, i.e., probability previous operation v_j ended under the condition it has begun. In this case, j is always unique preceding operation; This is the same expression as for other operations in SeqSeq;

$p^{fne} = 1$.

9. For finish operation a^f of FinJoin type p^{fb} counted the same way as for the JoinSeq; For both SeqFin and SeqJoin $p^{fne} = 1$; $p^{fa} = p_f^b p_f^{ne}$, just like for any other operation associated with any vertex.

IV. PROCEDURE TO COMPUTE PROBABILITIES OF CUTS AND CORRESPONDING SYSTEM STATES ACTUALIZATION

A. *Conditional Actualization of the Cuts Subsets*

Events of actions actualization generally shall be evaluated under the condition of other possible actions actualization at the same moment. This conditional dependence is modeled as conditional dependence in network cuts.

Actions in the network cut conditional actualization caused by network structure fragments types.

1) Types of Networks Structures Regarding Possible Networks Cuts: We consider two major types of network structure fragments in cuts, conditional cut sets (C) of vertices and non-conditional (N) subsets of vertices in the given cut. Conditional structures consist of more than one vertex and may include two types of network structures, waits sets (CW) and fork children (siblings) sets (CS).

CW are types of subsets CW_w of waiting actions (waits) in the given cut which have one common join—few such subsets in one cut possible.

CS are types of subsets $Cs_f, f \in F$ of the given cut, which consist of children or further descendant vertices of the common fork. Few such subsets in one cut are possible.

CJ are types of subsets Cj_j of join vertices in the given cut.

N- type subset (i.e., all other vertices are considered).

2) General Dependencies Cut Actualization: Actualization possibility for cut $P^{ca}(T_m)$ at the moment T_m takes form of:

$P^{ca}(T_m) = P^{cb}(T_m)P^{cne}(T_m)$;

$P^{cb}(T_m)-$ adjusted joint conditional cumulative distribution of given cut C vertices beginning at moment T_m;

$P^{cne}(T_m) = 1 - P^{ce}(T_m)$

$P^{ce}(T_m)-$ adjusted joint conditional cumulative distribution of given cut C vertices end (finish) at moment T_m;

$P^{cb}(T_m)$ depends on types N, CFS, CJ.

B. Dependencies for Cut Beginning Actualization

$P^{cb}(T_m) = P^{cbN*}(T_m)P^{cbJ}(T_m)P^{cbS}(T_m)$,

where $P^{cbN*}, P^{cbJ}, P^{cbS}$ calculated according each type of corresponding vertices sets (i.e., N, CJ, CS).

Note W type vertices are not distinguished when calculating $P^{cb}(T_m)$, and so, they considered to be N type in this calculation (but they will be distinguished when cut actualization end computed).

C. Algorithm to compute set of subnetwork decomposition onto the set of not intersected sets of vertices

Algorithm return set CB^*, used to compute probabilities of actualization beginning for set of vertices of N type. It is listed below.

Input: Vertex or set of independent vertices (cut, sub-cut) of the given network.
Output: Set CB of vertices of chains in the subnetwork of vertices preceding given ones, which has not pairwise intersections by vertices (and arcs).
initialization
create subnetwork of the current network which include given (terminal) vertices as last ones (and vertices, which reached on the way to these terminal vertices from the start)
Set list of visited vertices and visited arcs to NULL
while *not visited vertices in cut exist* **do**
  while *previous NOT visited vertices for this chain exist* **do**
    find the first arc with not yet visited first (topologically) start vertice
    create a new current empty chain in the list of chains
    set the first vertex of current empty chain to just found not visited vertice of not visited arc
    set this vertice as current
    add this vertice to the list of visited as starting (first) vertice
    // Find all further vertices in the current chain
    get first previous vertex of current vertex which is NOT in visited
    write visited arc to visited arcs
    write its number to visited vertices add a vertex to possible chain
  end
end
return Set CB^*

D. Conditional probabilities of cuts parts beginning depending cut vertices sets types

For N type vertices of each cut C:

$P^{cbN*}(T_m) = \prod_{i \in CB^*} F_i^b(T_m)$, where CB^*- set, given by algorithm suggested. It includes previous to set N vertices with wait type vertices CW which precede N (but not in C_h) excluded from CB and types N of C_h vertices added:

$CB^* = (CB \setminus CW) \cup N$.

CFS type. Subset with a few actions of the same fork children in one cut. Few CF subsets may happen in one cut,

In this case, same fork case (following same fork start) and so vertices in each subset CF of the given cut actions actualization in the cut depends functionally on other fork actions of the same fork start because their beginning is always happens in the same moment of one chosen fork action end.

Thus for each CF type subset Cf_f of the given cut,

$P_f^{cfb} = p_j^b \prod_{i \in CF_f} p_i^{b*}, j \in CF_f, i \in CF_f$.

p_j^{bj} is computed as a common part of all CF type vertices of one CF type set, before fork vertex v_j, such that these vertices precede this particular set of CF type.

Any (but one) j can be used for p_j^b computation, i.e., j is representative of the set type CF. Because all of p_j^b for different v_j in the given Cf_f are equal.

$p_i^{b*} = 1.0$, if path from common vertex to v_i is 1 (for siblings of common fork),

$p_i^{b*} = \prod_{p \in PredChain[i]} F_p^b$, if path from common fork vertex to v_i is over 1,

$PredChain[i]-$ numbers of this path vertices.

Here, p_i^{b*} is the cumulative joint probability distribution of the beginnings of the vertices in chains from a common fork.

For different CS sets $s \in S$ probability $P^{cfb} = \prod_{s \in S} P_s^{cfb}$.

CJ type. For vertex v_j of CJ type, $P^{cjb} = p_j^{bj}$;

Here, $p_j^{bj}-$ joint cumulative probability of join type vertex beginning computed according to procedures of vertices actualization computation.

E. Conditional probabilities of cuts parts end depending cut vertices sets types

$P^{ce}(T_m)$ depends on type W:

$P^{ce}(T_m) = P^{ceW}(T_m)P^{ce\overline{W}}(T_m)$,

where:

$P^{ceW}(T_m)-$ conditional probability of wait types vertices ending at the moment T_m;

$P^{ce\overline{W}}(T_m)-$ conditional probability of NOT-wait (all other types) vertices ending at moment T_m; CW type. Few subsets $Cw_w, w \in W$ of CW type is possible in the given cut (one per each join).

In this case, wait action actualization in the cut depends on other wait actions of the same CW subset of this cut actualization's in such a way that wait ended once all waits of the subset of waits began, which corresponds to the same join (to the same wait end, or sister waits). Such waits in one set may end in any possible sequences.

Thus, ends of W type set vertices depends conditionally on other waits and vertices previous to waits.

This dependence takes the following form:

$P_w^{cew} = 1$, iff all possible waits are in the given CW subset and so, $P_w^{cnew} = 0$, cut probability effectively equal to 0.

Note once all waits happen, then the next vertex began immediately.

$P_w^{cew} = 0$, otherwise, i.e. NOT all possible waits are in the given CW subset and so, $P_w^{cnew} = 1$.

This is because if NOT all waits happen in a given cut, then some previous vertices should. And this means neither one wait of the given set CW may end till condition previous vertex actual not changed (and other cut happened).

F. Probability Distribution of the Cuts actualization at the Given Moment

For the set $C(N_u)$ of possible cuts of the given $u-$th network N_u actualization probabilities of $C_h\ inC(N_u)$ form probability distribution:

$$\sum_{C_h \in C(N_u)} P_h^a(C_h) = 1.$$

Suggested procedures allow us to compute probabilities of any given cut $C_h \in C(N_u)$ of the network N_u and cuts probabilities distributions.

Based on this distribution stochastic process model of the system functioning according to action network, N_u can be built.

G. Algorithm to compute sets of network vertices with needed properties based on given prepared arrays

We consider Variation of the algorithm with a vector of vertices $Pr[i]$ preceding each current vertex $v_i \in C$ of the current cut C given. This vector is part of a global structure $U_{[u]}$ for the research of measures $\Omega_{[u]}$.

Here we consider cut C mentioned before as one of a kind C_h in a set of all cuts $C(N_u)$. To compute probability P_h^c of the given cut C_h we need to get its decomposition onto four sets:

$C_h = N \cup W \cup J \cup S$,

where N is a set of vertices, which probabilities of the beginning of the end are non - conditional;

$W-$ set of waits used to compute the probability of cut end;

$J-$ set of joins used to compute the probability of the beginning of the cut;

$S-$ set of siblings and their successors used to compute the probability of the beginning of the cut.

N set used to determine CB^*- set, given by algorithm above, with wait type vertices CW which precede C_h (but not in C_h) excluded from CB and current cut vertices added:

$CB^* = (CB \setminus CW) \cup C$.

Algorithm to find CB^* based on $Pr[i]$ suggested below:

```
Input: Vector Pr[i] for each v_i in C
Output: Set CB* for cut C
initialization
while int i  in C do
    while int k ≠ length(Pr[i]) do
        if (Pr[k][i] NotWait) AND ((Pr[k][i]) NOTIN (CB*)
          ) then
            ADD Pr[k][i] to CB*
        end
        k++;
    end
    i++;
end
CB*
```

V. Results of computations

Computations, according to provided procedures and algorithms, were implemented with JavaScript and with HTML for representation.

Results are shown at Fig. 7 and 8.

Results obtained allows to build and compute probabilities of cuts of alternative stochastic networks for any given moment. Each cut corresponds to the possible state of the system at the given moment. Cuts probabilities form the distribution of probabilities and allow them to compute distributions of different characteristics of states. Distribution of state characteristics at the given moment and cuts probabilities distributions allow the computing state of the corresponding stochastic process of the system functioning for each moment.

работы	ca	cb	ta	tb	tas	tbs	taf	tbf	-∞.-2.0	0.2.3	3.5.6	6.8.9	9.12.15	15.17.18	18.21.24	24.29.∞	prev	next	prevseq	vtypes	работы	сеть
0	0	0	5	5	-5	-5	0	0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000		1		STARTseq	**0**	**Fb**
1	0	0	3	6	0	0	3	6	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0	2,3	0	SEQfork	**1**	
2	0	0	3	9	3	6	6	15	0.0000	0.0000	0.9079	1.0000	1.0000	1.0000	1.0000	1.0000	1	_4	0,1	SEQ	**2**	
3	0	0	3	12	3	6	6	18	0.0000	0.0000	0.9079	1.0000	1.0000	1.0000	1.0000	1.0000	1	_5	0,1	SEQ	**3**	
_4	0	0	0	0	6	15	6	15	0.0000	0.0000	0.0000	0.1683	0.9079	1.0000	1.0000	1.0000	2	6	0,1,2	WAIT	**_4**	
_5	0	0	0	0	6	18	6	18	0.0000	0.0000	0.0000	0.1014	0.6852	0.9964	1.0000	1.0000	3	6	0,1,3	WAIT	**_5**	
6	0	0	3	6	6	18	9	24	0.0000	0.0000	0.0000	0.1014	0.6852	0.9964	1.0000	1.0000	_4,_5	k		JOINseq	**6**	
k	0	0	0	0	9	24	9	24	0.0000	0.0000	0.0000	0.0000	0.1393	0.7427	0.9781	1.0000	6		6	seqFINISH	**k**	
0	0	0	5	5	-5	-5	0	0	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		1		STARTseq	**0**	**pne**
1	0	0	3	6	0	0	3	6	1.0000	1.0000	0.0921	0.0000	0.0000	0.0000	0.0000	0.0000	0	2,3	0	SEQfork	**1**	
2	0	0	3	9	3	6	6	15	1.0000	1.0000	1.0000	0.8317	0.0921	0.0000	0.0000	0.0000	1	_4	0,1	SEQ	**2**	
3	0	0	3	12	3	6	6	18	1.0000	1.0000	1.0000	0.8986	0.3148	0.0036	0.0000	0.0000	1	_5	0,1	SEQ	**3**	
_4	0	0	0	0	6	15	6	15	1.0000	1.0000	1.0000	0.8986	0.3148	0.0036	0.0000	0.0000	2	6	0,1,2	WAIT	**_4**	
_5	0	0	0	0	6	18	6	18	1.0000	1.0000	1.0000	0.8317	0.0921	0.0000	0.0000	0.0000	3	6	0,1,3	WAIT	**_5**	
6	0	0	3	6	6	18	9	24	1.0000	1.0000	1.0000	1.0000	0.8607	0.2573	0.0219	0.0000	_4,_5	k		JOINseq	**6**	
k	0	0	0	0	9	24	9	24	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	6		6	seqFINISH	**k**	

Fig. 7. Actions actualisation Probabilities Computation Example

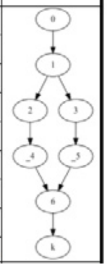
0	0	0	5	5	-5	-5	0	0	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		1		STARTseq	**0**	**pa**
1	0	0	3	6	0	0	3	6	0.0000	1.0000	0.0921	0.0000	0.0000	0.0000	0.0000	0.0000	0	2,3	0	SEQfork	**1**	
2	0	0	3	9	3	6	6	15	0.0000	0.0000	0.9079	0.8317	0.0921	0.0000	0.0000	0.0000	1	_4	0,1	SEQ	**2**	
3	0	0	3	12	3	6	6	18	0.0000	0.0000	0.9079	0.8986	0.3148	0.0036	0.0000	0.0000	1	_5	0,1	SEQ	**3**	
_4	0	0	0	0	6	15	6	15	0.0000	0.0000	0.0000	0.1513	0.2858	0.0036	0.0000	0.0000	2	6	0,1,2	WAIT	**_4**	
_5	0	0	0	0	6	18	6	18	0.0000	0.0000	0.0000	0.0843	0.0631	0.0000	0.0000	0.0000	3	6	0,1,3	WAIT	**_5**	
6	0	0	3	6	6	18	9	24	0.0000	0.0000	0.0000	0.0171	0.5355	0.2564	0.0219	0.0000	_4,_5	k		JOINseq	**6**	
k	0	0	0	0	9	24	9	24	0.0000	0.0000	0.0000	0.0000	0.0866	0.7400	0.9781	1.0000	6		6	seqFINISH	**k**	
0									1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000					**0**	
1									0.0000	1.0000	0.0921	0.0000	0.0000	0.0000	0.0000	0.0000					**1**	
2,3									0.0000	0.0000	0.9079	0.7474	0.0290	0.0000	0.0000	0.0000					**2,3**	
2,_5									0.0000	0.0000	0.0000	0.0843	0.0631	0.0000	0.0000	0.0000					**2,_5**	
3,_4									0.0000	0.0000	0.0000	0.1513	0.2858	0.0036	0.0000	0.0000					**3,_4**	
_4,_5									0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000					**_4,_5**	
6									0.0000	0.0000	0.0000	0.0171	0.5355	0.2564	0.0219	0.0000					**6**	
k									0.0000	0.0000	0.0000	0.0000	0.0954	0.7400	0.9781	1.0000					**k**	
cutssum									**1.0000**	**1.0000**	**1.0000**	**1.0001**	**1.0088**	**1.0000**	**1.0000**	**1.0000**					**cutssum**	

Fig. 8. Cuts actualisation Probabilities Computation Example

States of the system and states of the environment allow modeling information operations input and output. This further allows choosing further action network. If it is different from the previously chosen one, alternation happens, and the course of actions changed according to the new network. Such alternative stochastic networks chains form branches of the trees of possible alternative action networks and branches of alternative stochastic processes. Measurement of their correspondence to environment demands allows computing of the probabilistic measures of correspondence.

Models suggested allows us to estimate the multidimensional probabilistic measure $\hat{\Omega}(T_u)$, suggested in previous paper:

$$\hat{\Omega}(T_u) = \{(P(\hat{A}_u), P(\hat{E}_u), P(\hat{B}_{ui}), I_u^{er*} = \overline{1, |U|}, T_u = \overline{1, |U|}\}, \tag{1}$$

where the indexes run through all possible T_u, $pi_u^e \in \Pi_u^e$ and all possible u in $\hat{A}_{ui}$ i.e., through all possible dimensions of the complex index U which corresponds to probabilistic measure changes.

$S_i^e(i_i, pi_u^e)/\hat{E}_i-$ the realization of the environment state at moment T_i as a result of an alternative fulfillment $i_i \in I_i$ (result of event $\hat{E}_{ui}$);

$\hat{B}_{ui}(i_i, pi_u^s, \pi_u^e)/\hat{A}_i)-$ the event that due to an alternative fulfillment $i_i \in I_i$ state $S_i^s(i_i, pi_u^s, \pi_u^e)/\hat{A}_i$ will correspond to state $S_i^e(i_i, pi_u^e)$;

$\hat{A}_i$ according to description $\mathcal{R}$ of correspondence, given by the probabilistic correspondence predicate $p(S_i^s, S_i^e, i_i; \mathcal{R})$ (a twice vague probabilistic predicate):

$$P(\hat{B}_{ui}(i_i, \pi_u^s, \pi_u^e)/\hat{A}_{ui})) = Poss(p(S_i^s, S_i^e, i_i; \mathcal{R})); \tag{2}$$

As a result, scalar indicator of dynamic capability can be estimated:

$$\psi(O, C, S, M) := \sum_{i=1,u}^{i=I_u,u} \prod_{i=1}^{i=|I_u|} P(\hat{E}_{ui}) P(\hat{A}_{ui}) P(\hat{B}_{ui}), \tag{3}$$

which is the system's scalar capability indicator, a probabilistic measure value in $[0, 1]$ as it was shown at our previous publication

VI. Changing Functioning Process Mining Case

In general, the problem of alternative functioning process mining and alternative process research exist concerning technologies used for such alternative processes. We propose to use models that take the general form of nested graph-theoretic models.

Each realization of the system functioning under a given sequence of changing conditions modeled as complex graph-theoretic model G_r. G_r is complex because it is constructed with a few graph-theoretic models and relations between them.

Such realization of the system functioning may be considered as the set of traces for the given environment changes and impacts. Here trace corresponds to the same term in the process mining techniques.

It is ordered set $\mathbf{N}_{[u]}$ of probabilistic action networks $\mathbf{N}_{[u]}$: $\mathbf{N}_{[u]} = \langle N_{[u]} \rangle$, i.e. one of the dimensions of $U_{[u]}$ multidimensional array or list is numbers of networks $N_{[u]}$ in the sequences.

By $U[u]$ it can be understood as a universal index structure for the complex graph. Such an index structure is a multidimensional array or list, such that it contains numbers of all graph-theoretic objects mentioned in procedures for network vertices and cuts computation.

Example of this structure parts are shown at the Fig. 7 and 8 where structure is represented with HTML tables.

Functions $Fn^b(S), Fn^f(C)$ are defined by the technology of system functioning, including information technology.

Such functions define alternations cases.

Functions $Fn^b(S, T_m) : (S, T_m) \rightarrow N_{[u]}$ maps state $S(T_m)$ of the system $S^s(T_m)$and its environment $S^e(T_m)$ at the moment T_m to the chosen network $N_{[u]}$ of operations $a_{[u]} \in A_{[u]}$ and their characteristics $Ch(a_{[u]})$.

Each $N_{[u]}$ corresponds to $\langle S^b_{N_{[u]}}, S^e_{N_{[u]}}, \{C_{N_{[u]}}\} \rangle$ of network beginning, end states and states of cut $C_{N_{[u]}}$ realization.

$\mathbf{G}_y, y \in Y, Y \in U$. Alternatives $\mathbf{G}_u, u \in U$ of models $\mathbf{G}_y$ are the models of states and transitions (probably, nested and complex ones), which are built for certain alternative technologies of the system $\mathbf{T}^s$ and environment $\mathbf{T}^e$ functioning. States and transitions at models determine branches of trees. Each tree corresponds to an alternative set of branches. Each branch corresponds to an alternative sequence of states and transitions resulting from the cause - and - effect relationships. These relationships link different types of states, including information ones. Structure of $\mathbf{G}_y$ defined by alternative decisions, the structure of $\mathbf{G}_u$ defined by characteristics of technologies used (including IT) and environment states. Probabilities $\mathbf{p} := \langle \mathbf{p}_u(\mathbf{G}_u), \mathbf{p}^d_u(\mathbf{G}_u), \mathbf{p} \rangle$ of transitions and states characteristics $\mathbf{C}_u, \mathbf{C}^d_u$ are known. They are defined by technologies: $\mathbf{T}^s-$ used by the system and $\mathbf{T}^e-$ used by its environment.

Let us designate Log the structure of log- files that describe families of alternative stochastic action networks.

Than, the objective of alternative processes discovery is obtaining $Log \rightarrow \{FASAN\}$ mapping, alternative process model preparation is $FASAN \rightarrow U[u]$ mapping, than $U[u] \rightarrow \Omega_{[u]}$ is capabilities indicators $\Omega_{[u]}$ estimation mapping and $\Omega_{[u]} \rightarrow \Pi$ is alternative processes enhancement Π mapping.

Apart of this sequence of mappings, $U[u] \rightarrow \Delta(Log^*)$ is conformance $Delta$ to new data (Log^*) checking mapping.

Based on $Delta$ decision taken to update or learn $FASAN$ model, based on Ω decision taken to enhance alternative processes and technologies, including information operations.

VII. Conclusion

As a result of the suggested models and algorithms, the quantitative estimation of system capability and other operational (pragmatic) properties concerning changing environment and further information technology use becomes possible depending on the problems' parameters and variables to be solved. Among the features that can be estimated are dynamic capability, organizational capability, information technology use performance, and digitalization performance. Suggested models can be used for process mining of alternative functioning of systems in changing environments and for the research of information, operations performed to alternate such functioning. Such models, used for process science research, could further decide the alternative action requirements synthesis problems.

Acknowledgment

The reported study was funded by RFBR, project number 20-08-00649 and 19-08-00989.

References

[1] D. B. Laney, *Infonomics [electronic resource]: How to Monetize, Manage, and Measure Information as an Asset for Competitive Advantage*. Milton: Routledge, 2017.

[2] M. Pańkowska, *Infonomics for distributed business and decision-making environments: Creating information system ecology*. Hershey: Business Science Reference, 2010.

[3] K. Brand and H. Boonen, *IT Governance based on CobiT 4.1: A management guide*, 3rd ed., ser. Best practice. [Zaltbommel]: Van Haren, 2010.

[4] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[5] A. Geyda and I. Lysenko, "The complex of models for system capability estimation with regard to information technology use," in *AMCIS 2020 Proceedings*. Utah: AIS, 2020, vol. 6. [Online]. Available: https://aisel.aisnet.org/amcis2020/strategic_uses_it/strategic_uses_it/6

[6] N. Ahmad and J. Ribarsky, "Towards a Framework for Measuring the Digital Economy," in *16th Conference of the International Association of Official Statisticians (IAOS) OECD Headquarters, Paris, France*. Paris, France: IAOS, 19-21 September 2018.

[7] F. Comim, M. Qizilbash, and S. Alkire, *The capability approach: Concepts, measures and applications*. Cambridge and New York: Cambridge University Press, dr. 2014.

[8] M. Carmona and L. Sieh, *Measuring quality in planning: Managing the performance process / Matthew Carmona and Louie Sieh*. London: Spon, 2004.

[9] H.-D. Kochs, *System Dependability Evaluation Including S-dependency and Uncertainty*. Cham: Springer International Publishing, 2018.

[10] A. Geyda and I. Lysenko, "Modeling of Information Operations Effects: Technological Systems Example," *Future Internet*, vol. 11, no. 3, p. 62, 2019.

[11] L. Portinale and D. C. Raiteri, *Modeling and analysis of dependable systems: A probabilistic graphical model perspective*. New Jersey and London and Singapore and Beijing and Shanghai and Hong Kong and Taipei and Chennai: World Scientific, 2015.

[12] A. Geyda, "Models and Methods of Optimal Information Operations Use for System Functioning," in *Proceedings of the 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019)*. Paris, France: Atlantis Press, 2019, pp. 15–22.

[13] A. S. Geyda and I. V. Lysenko, "Information Technology Efficiency models for Agile system's functioning," in *Conference of Open Innovation Association FRUCT*. Finland: FRUCT Oy, 2018, vol. 22, pp. 313–319.

[14] A. Ustundag and E. Cevikcan, *Industry 4.0: Managing The Digital Transformation*. Berlin: Springer International Publishing, 2018.

[15] OECD, *Digitalisation and productivity: A story of complementarities*. Organisation for Economic Co-Operation and Development (OECD), 2019.

[16] ——, *Roadmap: The digitalisation of science*. Organisation for Economic Co-Operation and Development (OECD), 2019.

[17] ——, *Business dynamics and digitalisation*. Paris,France: Organisation for Economic Cooperation and Development (OECD), 2019.

[18] W. Kuklys, *Amartya Sen's capability approach: Theoretical insights and empirical applications*, ser. Studies in choice and welfare. Berlin: Springer, op. 2010.

[19] S. Deneulin, M. Nebel, and N. Sagovsky, *Transforming unjust structures: The capability approach / edited by Severine Deneulin, Mathias Nebel and Nicholas Sagovsky*, ser. Library of ethics and applied philosophy v. 19. Dordrecht and London: Springer, 2006. [Online]. Available: http://www.springer.com/gb/BLDSS

[20] T. Fehlmann, "Computer Science and Digitalization," *Athens Journal of Sciences*, vol. 5, no. 3, pp. 247–260, 2018.

[21] V. Parida, D. Sjodin, and W. Reim, "Reviewing Literature on Digitalization, Business Model Innovation, and Sustainable Industry: Past Achievements and Future Promises," *Sustainability*, vol. 11, no. 2, p. 391, 2019.

[22] A. S. Geyda, "Conceptual and Formal Models of Usage Effects of Information Operations in Technological Systems," in *Proceedings of the 24TH FRUCT conference*, Balandin S., Ed. Bologna, Italy: FRUCT Oy, 2019, pp. 599–607.

[23] ——, "Models and methods to estimate digitalization success predictively," in *Workshop on computer science and information technologies*, Yousoupova N.I., Ed. SPIIRAS, 2019, vol. 2019.

[24] A. Geida, "The modeling in the course of technical systems investigation: some expansions of the graph theory usage," *SPIIRAS Proceedings*, vol. 2, no. 17, pp. 234–245, 1 1. [Online]. Available: http://proceedings.spiiras.nw.ru/index.php/sp/article/view/1553

[25] D. Golenko-Ginzburg and A. Gonik, "Project Planning and Control by Stochastic Network Models," in *Managing and Modelling Complex Projects*, T. M. Williams, Ed. Dordrecht: Springer Netherlands, 1997, pp. 21–45.