Model for Verification of Intelligence of Multiagent Systems

Oleksandr Martynyuk[a], Oleksandr Drozd[b], Anatoliy Sachenko[c], Kostiantyn Zashcholkin[d], Igor Mileiko[e]

[a,b,d,e] *Odessa National Polytechnic University, Ave. Shevchenko 1, 65044 Odessa, Ukraine,*
[c]*Ternopil National Economic University, Victory Square 3, building 2, 46020, Ternopil, Ukraine*

Abstract

Modern distributed information systems are characterized by the integration of the properties of advanced network technologies, including Web services, Clouds, "Green" and other technologies. In most of them, the intelligence, inherent in multi-agent systems, is becoming more and more important. Intelligence, as a set of properties of knowledge and mechanisms of their formation, is characterized by special solutions. This specialization is reflected in all tasks of systems synthesis and analysis, in particular, tasks of their verification and testing. The model for verifying the intelligence of multi-agent systems has been further developed in this paper. The model based on behavioral automata experiments and extended Petri nets. The features of the proposed verification model are the hierarchy of goals, the composite knowledge model and inference mechanism, the identification of behavioral properties. Functional specifications of behavioral models of multi-agent systems with properties and mechanisms of intelligence define sets and relationships for atomic events and actions, which represent static properties and mechanisms of multi-agent systems, in positions and transitions of Petri nets. The specifications also define sets and relationships of tokens, which are moved in positions and transitions and represent the dynamic properties and mechanisms of multi-agent systems. The model makes it possible to determine the necessary conditions for special methods of verification and testing of intelligence, as properties of distributed information systems, and to reduce their computational complexity. The overall computational complexity of the *k*-decomposition for models of verification of intelligence on base of hierarchical Petri nets is reduced exponentially by a factor of *k*, and the experimental verification, which was held for real MAC, confirm this decrease.

Keywords

Distributed Information System; Knowledge Model; Behavior Verification; Verification Model; Property Identification

1. Introduction

The development and penetration of modern distributed information systems (DIS) [1] into various areas of human activity is accompanied by the use of a variety of computer technologies and platforms. These technologies are based on the latest achievements, on the one hand, in mathematics and physics of storage, transmission and processing of information, design and production of technical components and systems, on the other hand, in the organization of special processes in different application areas and for solving various applied problems. In fact, emerging complex DIS is characterized by the integration of many architectural, structural-functional, information, methodological properties and mechanisms of progressive technologies, including Web-services, P2P, Clouds, GRID, Clusters and Multi-Agent [2-4], communication Transport-Network (including Wireless) and "Green" [5], also many other technologies. Most modern technologies are increasingly mastering and transferring to DIS the ability of their own goal-setting and intelligence [6]. Together with intelligence, other new properties and mechanisms also develop and become frequent and even integral, in particular, the autonomy and mobility [7] of the components and subsystems

IntelITSIS’2022: 3nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23–25, 2023, Khmelnytskyi, Ukraine
EMAIL: anmartynyuk@ukr.net (O. Martynyuk); drozd@ukr.net (O. Drozd); as@wunu.edu.ua (A. Sachenko); const-z@te.net.ua (K. Zashcholkin); mig3@ukr.net (I. Mileiko)
ORCID: 0000-0003-1461-2000 (O. Martynyuk); 0000-0003-2191-6758 (O. Drozd); 0000-0002-0907-3682 (A. Sachenko); 0000-0003-0427-9005 (K. Zashcholkin); 0000-0002-8664-7037 (I. Mileiko)

of DIS,the dynamic cooperativity [8] of their functioning. The last three properties, together with intelligence, form a system of distinctive features of Multi-Agent Systems (MAS) [9], in connection with which many DIS explicitly or implicitly integrate MAS, specialized for them.

Trustworthiness of operation of DIS, which are highly complex systems as part of various hardware, communication, software, information tools, is possible only, if they have sufficient values of error-free, reliable functioning, and operability [10]. A high level of these characteristics is ensured by many hardware, functional, informational, methodological tools of verification, online and offline testing [11], performed respectively at the levels of projects and implementations of DIS, often integrated into DIS-specific automated systems of technical diagnostic [12]. At the same time, it is the ultra-high complexity of modern DIS shifts their formal, design and operational specifications, which are necessary for verification and testing to the upper, system-architectural, structural-functional levels [13]. This complexity more and more leaves the logic-circuitry and design-technological levels [14] for manufacturing specifications and component debugging of DIS. It follows that the completeness, accuracy and relevance of verification and testing at the upper levels to a decisive extent depends on the formal specifications of the behavior of DIS [15], mandatory for these levels, the means and capabilities of the analysis of these specifications. Such a behavioral analysis, as a rule, is NP-complex [16], it concerns, first of all, formal specifications, which are based on models and methods of an automata class [17]. This fact prompts the use of various types of spatial-temporal [18] and functional [19] decomposition of automata models and methods, as well as their general improvement and specialization [20] according to specific features of DIS and their components. Intelligence, as a set of properties of knowledge and mechanisms for their analysis and derivation, is characterized by special solutions [21]. Such specialization is reflected in all tasks of analysis and synthesis of intelligent distributed information systems, in particular, tasks of their verification and testing. This circumstance can be essentially used in the functional decomposition of DIS [22], highlighting the property of intelligence and accentuating it in special extended Petri nets, which represent the class of automata class models most effective for behavioral analysis and are used in behavioral verification and testing of DIS.

2. Goal and Tasks

The goal of creating a new verification model for intelligence of MAS is to reduce the computational complexity and increase the completeness of verification for the purposeful intellectual behavior of DIS.

To achieve this goal the tasks of constructing formal models of agents and MAS in general, which are based on extended hierarchical Petri nets, are solved for: a) formal input specifications; b) hierarchies of dynamic objective and verification goals and subordinate tasks; c) composite models of subject knowledge and mechanisms for their derivation for fragments of behavior and achievable structures; d) verification of behavior based on automata check and recognition experiments; e) composite knowledge models and mechanisms for inference for verifying fragments of behavior and achievable structures for verifying behavior.

The constructed models make it possible to determine the formal conditions and basic steps for a set of methods for behavioral verification of the properties and mechanisms of MAS intelligence. Each of these methods can differ in the depth of knowledge retention, dependence and parallelism of actions, target parameters and optimization procedure, when achieving results.

3. Model of Intelligence for Verification

3.1. Local energy properties and instant energy consumption state diagrams

Formal functional specifications of behavioral models for MAS, which contain properties and mechanisms of intelligence, is formed on the basis of extended hierarchical Petri nets. Such specifications define sets and relations for atomic conditions, events as condition structures, actions as function structures, and atomic functions, that represent static properties and mechanisms of the MAC in positions and transitions of Petri nets. The Petri nets also define the sets and relationships of tokens and their structures, that move in positions and transitions during its operation, which represent the dynamic properties and mechanisms of the MAS. The reconfigurable extended hierarchical Petri net looks like [23]:

$$S(f) = (P, T, Ev, Ac, X, Y, In, Pr, Fs, Fd, M, fM_0). \quad (1)$$

Here, in addition to the usual components of Petri net, extensions are represented by sets of events *Ev*, actions *Ac*, conditions *X*, functions *Y*, time intervals *In*, probability-priorities *Pr*, essential for intelligence, *M* – is the set of token-labels of different n_m types:

$$Fs:(((B(P\times Ev)\times Pr\times Int)\rightarrow((B(P\times Ev)\times Pr\times Int)\times(T\times Ac\times Pr\times Int)))\cup((T\times Ac\times Pr\times Int)\rightarrow$$
$$\rightarrow((T\times Ac\times Pr\times Int)\times(B(P\times Ev)\times Pr\times Int)))$$

The energy consumption for the minimum, average and maximum reference values of the reduced numbers of one- and multi-circle switching of the events, that have occurred in positions, and actions, performed in the transitions, is localized in the components of the DIS and can be represented in the form of instantaneous energy structures and their compositions based on the spatial architecture DIS.

So, on the basis of the entire Petri net $S(f)^E$ or some Petri subnet for some of its special components, as their sub-models, diagrams of attainable instantaneous (at a certain moment of time *t*) energy states or markings (sections), marking the above events can be formed positions and transition actions, type:

$$Sp_t=\{(p_{i1},ep_{i1}),(p_{i2},ep_{i2}),\ldots,(p_{ik'},ep_{ik'}),\ldots,(p_{ik},ep_{ik})\}\cong\{(x_{i1},ep_{i1}),(x_{i2},ep_{i2}),\ldots,(x_{ik'},ep_{ik'}),\ldots,(x_{ik},ep_{ik})\}$$
$$St_t=\{(t_{j1},et_{j1}),(t_{j2},et_{j2}),\ldots,(t_{jk'},et_{jk'}),\ldots,(t_{jk},et_{jk})\}\cong\{(y_{j1},et_{j1}),(y_{j2},et_{j2}),\ldots,(y_{jk'},et_{jk'}),\ldots,(y_{jk},et_{jk})\} \quad (2)$$

– the static relation of incidence expectations of subsets of positions from the boolean $\{p_1,\ldots,p_{ip}\}\in B(P)$ and transitions from *T*, which depends on the values of the variables of current events, actions, probabilities/priorities and intervals,

$$Fd:(((B(P\times Ev\times M^{\lfloor nm \rfloor})\times Pr\times Int)\rightarrow(B(P\times Ev)\times Pr\times Int)\times(T\times Ac\times M^{\lfloor nm \rfloor}\times Pr\times Int)))\cup$$
$$\cup((T\times Ac\times M^{\lfloor nm \rfloor}\times Pr\times Int)\rightarrow((T\times Ac\times Pr\times Int)\times(B(P\times Ev\times M^{\lfloor nm \rfloor})\times Pr\times Int)))$$

– dynamic relation of activation of incidents of subsets of positions from the Boolean $\{p_1,\ldots,p_{ip}\}\in B(P)$ and transitions from *T*, which, in contrast to the static relation *Fs*, also depends on the markup values *M*, upon activation, causes modification of the values of variables of events, actions, probabilities/priorities, intervals and markup with changing and saving their new values. Here $M^{\lfloor nm \rfloor} = B(\cup_{im=0}^{nm}\{M^{im}\}$ – is the space of sets of token labels placed in positions M^{im} –is the direct Dekart i_n-degree of the set *M*.

By construction, the space-time structure and the agent-based, hierarchical functional decomposition of the extended Petri net for the MAS corresponds to the network, multi-level and functional structure of MAS and defines Petri nets for the selected structure of the behavioral model of the MAS of a certain level with the properties and mechanisms of intelligence. Decomposition performs the grouping of components and subnets for agents and functions in MAC model. As a result, on the basis of Petri net, a set of network and hierarchical models is obtained [22, 23] with the subnets-functions allocated for the analysis of intelligence:

$$nS = (X, Y, S(f)^S, \alpha),$$
$$SI = (S(f), \cup_{i\in I}S(f)_i^p, \cup_{j\in J}S(f)_j^t, Sg_{iS}) \quad (3)$$

Special, general and detailed models of agents [24]

$$ag_i=(ag_{Ri}, ag_{Di}, Ev_i, Ac_i, \Delta_i),$$
$$ag_i=(cS_i, cM_i, Q_i, St_i, \{o_i, \sigma_i, \alpha_i\}, \{cs_{0i}, cm_{0i}, q_{0i}, st_{0i}\}), \quad (4)$$

determine the level of intelligence, reactive ag_{Ri} or deliberative ag_{Di}, actions Ev_i, events Ac_i, special models of behavior analysis (in particular, verification) cS_i ($S(f)_i$ includes into cS_i,, as a component), special methods for analyzing behavior cM_i, goals Q_i, strategy tasks St_i, special agent-based mechanisms $\{o_i, \sigma_i, \alpha_i\}$ observation, application of strategy, adaptation, initial agent-based control model, method, goal, strategy $\{cs_{0i}, cm_{0i}, q_{0i}, st_{0i}\}$.

The models ag_i make it possible to define in its composition a formal model of intelligence of MAS and, after its functional decomposition and reduction, build a behavioral verification model for it. The necessary conditions are formed for a group of possible methods for verifying the intelligence of MAS.

3.2. Model of Goals and Tasks

Subject intelligence in the MAS can be localized in some agents, have functional completeness within the selected agent, specialized in a certain set of goals, tasks, properties and functions. In another case, subject intelligence can be distributed between agents of different types, for example, by the functions: a) storage of knowledge, inference, transport; b) steps of evolutionary optimization, synchronization of steps, branches and optimization iterations. Along with one subject intelligence, there may be another one, that differs from the first in terms of the totality of subject tasks, in particular, separating the MAS resources from the first.

It should be noted the existence of hierarchies – the goal $q_i \in Q_i$ defines the tasks-strategies St_i, each $st_i \in St_i$ of which, in turn, determines the events Ev_i (into them and conditions X_i), actions Ac_i (into them and functions Y_i), necessary for the solution, as well as the resources required for them. Moreover, in this case, the division is obvious between the goals Q_i of some common tasks St_i, between the tasks – some common events Ev_i and actions Ac_i with their resources. That is, the hierarchy is present as a spanning tree in such a multi-level network structure of goals Q_i, tasks St_i, events Ev_i and actions Ac_i with their resources.

In any case, depending on the organization of the subject MAS, the general model of intelligence, like the model of cooperativity [25] by Petri nets of MAC (Petri net for agent - on the Figure 1), can have a distributed basic peer-to-peer (linear, ring, network), hierarchical or combined structure.

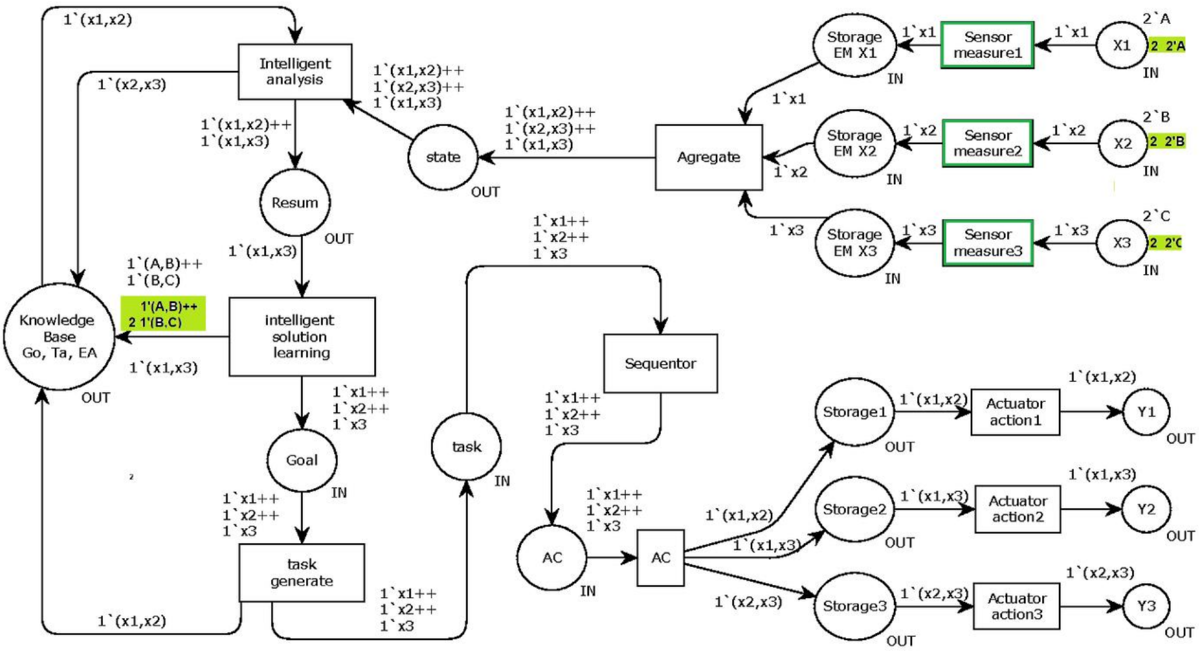

Figure 1: Petri net model for energy agent

Several levels in the structure are often typical for MAS with a fixed, static hierarchical structure of connections of subject weakly reconfigurable goals Q_i and tasks St_i for stable agents Ag_i of predefined types of different levels, although the agents can change goals and tasks, passing from one type to another.

Peer-to-peer structures of the intelligence model are applied, firstly, as a degenerate case of multi-rank structures; secondly, as the initial states of dynamic multi-rank structures with a situational reconfigurable hierarchy of goals Q_i and tasks St_i of individually reconfigurable agents Ag_i. In the process of functioning in reconfigurable hierarchies, agents can appear, disappear, move from one type and rank to another, the types and ranks of the hierarchy themselves can also arise and disappear dynamically, situationally.

This work considers analyze of peer-to-peer and two-rank architectures, to which it is possible to reduce multi-rank hierarchies without loss of generality of this analyze.

Specific goals Q_i in these static or dynamic structures of intelligence models of MAS for any level of the hierarchy are defined as some of the transformations attainable at this level $q_i=((cs_{i1}, cm_{i1}), (cs_{i2}, cm_{i2}))$: two "analysis model - analysis method" (cs_{i1}, cm_{i1}), initial for the goal, transformed into a target two "analysis model - analysis method" (cs_{i2}, cm_{i2}). Moreover, the analysis model cs_{i1} of the initial pair for the goal is characterized by the initial values of the subject formal target criteria, and the analysis model of the target pair is characterized by the final, target values of the subject formal target criteria. In verification of behavior of agent ag_i, the main criteria cR_i can be taken as its computational complexity κo_i, completeness ϕu_i, length λe_i, multiplicity θu_i and localization λo_i (within the input space-time structure of the MAS), additional criteria can be taken as controllability ρe_i (realizability), transportability $o b_i$ (observability), heritability ιn_i (within the same input space-time structure of the MAS), so $cR_i=(\kappa o_i, \phi u_i, \lambda e_i, \theta u_i, \lambda o_i, \rho e_i, o b_i, \iota n_i)$. Situationally, dynamically, some criteria of goals Q_i and tasks-strategies St_i of some agents ag_i for some time may go beyond the selected main criteria of general goals and tasks-strategies of the analysis of MAS. For example, to verify the intelligence of MAC, such criteria of goals of some agents, subordinate to the main ones, can be to ensure their controllability ρe_i, observability $o b_i$, and inheritance ιn_i, the tasks-strategies St_i, corresponding to these criteria, both tasks St_i as part of goals Q_i, – are the allocation of parameterized agent subnets in Petri nets of MAC with these criteria-parameters, that provide implementation, transportation, and inheritance of the necessary verifying behavior. That is, for individual agents, in addition to their main goals with formalized in criteria acceptable values of computational complexity κo_i, completeness ϕu_i, length ϕu_i, multiplicity θu_i, localization λo_i, it's possible the emergence of additional goals localized in space and time of the behavior of agents with formalized in criteria acceptable values of controllability ρe_i, observability $o b_i$, heritability ιn_i.

In this case, the goals Q through criteria cR of the models and the tasks-strategies St of the MAS of the higher level, which are long-term in the life cycle of the MAS, control the parametrized goals Q_i of the agents of the lower-adjacent level, which are shorter-term, dynamic in the life cycle of the agent and the MAS. The higher criteria cR control through the parameterized lower goals Q_i and their corresponding parameterized models cS_i, methods cM_i and tasks-strategy St_i of the corresponding lower-level agents. Thus, control is carried out as a parametric adjustment of the lower models cS_i, methods cM_i and tasks-strategies St_i, corresponding to the lower goals Q_i, the values of the parameters cR_i necessary for the upper tasks St. As noted above, for verification goal parameters can be the required values of computational complexity κo_i, completeness ϕu_i, length λe_i, multiplicity θu_i, localization λo_i, controllability ρe_i, observability $o b_i$. and inheritance ιn_i. When used as behavioral models of Petri nets, such parametrization characterizes the components, agent subnets $S(f)^S\wedge$, $\cup_{i\in I}S(f)_i^p$, $\cup_{j\in J}S(f)_j^t$, general Petri nets $S(f)$, which represent lower and higher models and tasks-strategies as part of the corresponding goals, and is implemented on the properties of specific positions P_i, transitions T_i, events Ev_i, actions Ac_i, movements of tokens M_i and their aggregates. The general scheme of actions for controlling subordinate lower agent models cS_i, methods cM_i and tasks-scenarios St_i using their goal parameters from the side of the higher goal parameters cR from the higher agent models cS, methods cM and task-scenarios St for the formal two-level model of the analysis of MAS include the definition and analysis of: a) the higher goal parameters cR and the corresponding tasks-scenarios St for their achievement; b) lower models cS_i, as submodels of the higher model cS,, lower methods cM_i and lower tasks-scenarios St_i; c) the relationship between the dependence of lower models cS_i, mrthods cM_i and tasks-scenarios St_i on the higher goal parameters cR in the corresponding task-scenarios St; d) mappings from the higher goal parameters cR and the corresponding their task-scenarios St to the lower goal parameters Q_i and the corresponding their task-scenarios St_i. General higher goals Q and subordinate lower goals Q_i place criteria-parameters cR, cR_i,, as dynamic parametric weight extensions of the general Petri net $S(f)$ of MAS and localized Petri subnets of its agents $S(f)^S\wedge$, $\cup_{i\in I}S(f)_i^p$, $\cup_{j\in J}S(f)_j^t$ in the formal models cS, cS_i and methods cM, cM_i of analyze of the behavior and tasks-scenarios of MAS St and its agents St_i.

As noted, each formally defined goal of the intelligence models ща MAS give its own set of tasks to be solved, that allow it to be achieved, and the tasks can be shared with other goals.

Each task-strategy $st_i\in St_i$ is defined as a scenario – asynchronous-event parallel procedure of application of previous analysis methods cM_{i1} to previous analysis models cS_{i1} from goal (the first in the goal pair of twos "previous model-method" - "next model-method" $q_i=((cs_{i1}, cm_{i1}), (cs_{i2}, cm_{i2})))$ – with defined start and finish, for example, formally represented as parallel graph-scheme of an algorithm with a single start and finish, functional nodes, possible branching nodes, loops, input/output, assuming transmission values of properties (data and parameters), functions (methods and handlers).

Tasks-strategies St, St_i and the methods cM, cM_i they call can assume deterministic, pseudo-random, exhaustive, intelligent analysis, synthesis, in particular, optimization, as well as when using special vertices – parallelism, recursiveness, and iteration. Such extended graph-schemes can serve as an initial, visual form for constructing equivalent (and, if necessary, more formalized) models for tasks-strategies St, St_i and methods cM, cM_i in their composition in the form of appropriate special analysis for models, based on extended hierarchical Petri nets nS, SI (3), which can be obtained independently from graph-schemes.

Analysis models cS_i and methods cM_i, which are based on extended hierarchical Petri nets nS, SI, together with the tasks-strategies St, St_i scenarios of the analysis of MAS and criteria cR_i, as part of the goal pair of twos “previous model-method” - “next model-method” $q_i=((cs_{i1}, cm_{i1}), (cs_{i2}, cm_{i2}))$, allow us to represent the upper levels of formal analysis of specifications for the behavior of MAS, including giving an external description and placement of their intelligence. Subsequent decomposition and detailing of these Petri nets at the lower levels of the formal multi-level model make it possible, in particular, to determine the internal realization of the intelligence of MAS, both for subject and verification tasks.

3.3. Knowledge Model and Intelligence Inference Mechanism

The intelligence of the MAS, limited by the resources of the DIS, and its handlers for reactive agents ag_{Ri}, when making decisions, most often one-step, are based on statistics, experience, including expert experience, that is, are based on simple models of accumulated knowledge. In this case, the MAS and agent knowledge bases are quite simple, change slowly, and can usually be localized in separate agents.

In the case of deliberative agents ag_{Di}, intelligence presupposes the possibility of obtaining a forecast of the consequences of decision-making in more than one step, that is, some more complex knowledge and forecasting model is needed, much more complex, than the knowledge model of reactive agents. Accordingly, large resources are required to be provided by DIS for such agents of MAS, and these resources are allocated usually in a MAS and DIS environment.

In the formal models of the knowledge space knS and knS_i presented below, for the MAS models and its agents, atomic, transactional data are understood as lexemes Lex^s, $\cup_{i\in I}Lex_i^s$ are minimal linear (without branching and feedback) fragments of the behavior of MAS models and its agents, its own semantics, weakly related to the subject semantics of the MAS, that is, a set of lexemes – the MAS vocabulary. Lexemes have their own specific subject and goal interpretation.

For example, for subject interpretation in MAS intelligence, the simplest one-step lexeme represents the formation of a production step $(A\to^{pr,int}B)_{Prod} \cong \sigma_i(A)^{pr,int}=B$ (here σ_i is the application of the step of the agent's strategy presented above, a step in in this case, production, $pr\in Pr$, $int\in Int$ - respectively the probabilities / priorities and time intervals presented earlier) or the union of two slots (one informational, the second functional) in the simplest frame $(Data_{sl1},Func_{sl2}{}^{Pr,In}()_{fr})_{fr} \cong ((Data_{sl1},\sigma_{sl2}{}^{pr,int}))_{fr}$ (here $\sigma_{sl2}{}^{pr,int}$ is the use of a frame step of the agent's strategy - one inference step).

For goal interpretation in MAS behavior, a one-step token assumes the presence of external references (inputs and outputs) for the productions $\to^{In}(A\to^{pr,int}B)_{Prod})^{Out}\to \cong \to^{In}(\sigma_i^{pr,int}(A)=B)^{Out}\to$ (here $\to^{In}$, ${}^{Out}\to$In, is the external appeal of logical inference to and from production, respectively) or the internal feasibility of the frames $Func_{sl2}{}^{Pr,In}(Data_{sl1})_{fr}\neq\varnothing \cong \sigma_i{}^{pr,int}(Data_{sl1})_{fr}\neq\varnothing$ (the ability to exit from it). In the representation of MAS behavior using Petri nets $S(f)$, $\cup_{i\in I}S(f)_i$ the set of static lexemes Lex^s, $\cup_{i\in I}Lex_i^s$ is a set of sequences of pairs of subsets of events and actions

$$Lex^s\in W=((B(Ev)\times Pr\times Int)\times(Ac\times Pr\times Int))^*,\ Lex_i^s\in W_i=((B(Ev_i)\times Pr_i\times Int_i)\times(Ac_i\times Pr_i\times Int_i))^*,$$

defined on the basis of static incidence relations Fs, $\cup_{i\in I}Fs_i$ and taken as indivisible (without branches), such that for each of them there are no others in this set that are less than the given one, they are a fragment of the given one. In these sequences for pairs of subsets of events for positions and actions in the transitions, incident to them, it is not defined in which they can be in the waiting, ready or activated.

For subject interpretation using Petri nets, the simplest one-step static tokens are represented:

a) for the production step in the form

$$\sigma_i^{pr1e,int1e}(((\cup_{i1\in I1}(ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a})))=((\cup_{i1'\in I1'}(ev_{i1'}),pr_{1'e},int_{1'e}),$$
$$(ac_{1'},pr_{1'a},int_{1'a}),(\cup_{i2\in I2}(ev_{i2}),pr_{2e},int_{2e})) \cong$$
$$Fs((\cup_{i1\in I1}(p_{i1},ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a})))=((\cup_{i1'\in I1'}(p_{i1'},ev_{i1'}),pr_{1'e},int_{1'e}),$$
$$(ac_{1'},pr_{1'a},int_{1'a}),(\cup_{i2\in I2}(p_{i2},ev_{i2}),pr_{2e},int_{2e}));$$

b) for the step of the simplest two-slot frames

$$((\cup_{i1\in I1}(p_{i1},ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a}))_{sl1},\sigma_{1sl2}{}^{pr,int})_{fr} \cong$$
$$\alpha(((\cup_{i1\in I1}(p_{i1},ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a}))_{sl1})_{fr},$$

here α is the application of the step of the strategy $\sigma_{1sl2}^{pr,int}$ of the agent as the identification of the support subset of positions / transitions $\cup_{i1\in I1}(p_{i1},ev_{i1})$. For target interpretation using Petri nets, the simplest one-step static tokens are represented: c) for products with external inputs and outputs

$$\rightarrow^{In}(\sigma_i^{pr1e,int1e}(((\cup_{i1\in I1}(ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a})))=((\cup_{i1'\in I1'}(ev_{i1'}),pr_{1'e},int_{1'e}),$$
$$(ac_{1'},pr_{1'a},int_{1'a}),(\cup_{i2\in I2}(ev_{i2}),pr_{2e},int_{2e})))^{Out}\rightarrow \cong$$
$$\rightarrow^{In}(Fs((\cup_{i1\in I1}(p_{i1},ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a})))=((\cup_{i1'\in I1'}(p_{i1'},ev_{i1'}),pr_{1'e},int_{1'e}),$$
$$(ac_{1'},pr_{1'a},int_{1'a}),(\cup_{i2\in I2}(p_{i2},ev_{i2}),pr_{2e},int_{2e}))^{Out}\rightarrow;$$

d) for a two-slot frame with the liveliness property

$$\sigma_{1sl2}^{pr,int}(((\cup_{i1\in I1}(p_{i1},ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a}))_{sl1})_{fr}\neq\varnothing \cong$$
$$\alpha(((\cup_{i1\in I1}(p_{i1},ev_{i1}),pr_{1e},int_{1e}),(ac_1,pr_{1a},int_{1a}))_{sl1})_{fr}\neq\varnothing.$$

Dynamic verifiable lexemes Lex^d, $\cup_{i\in I}Lex_i^d$ for Petri nets are formed from static tokens as their narrowing based on the application of dynamic incidence relations Fd, $\cup_{i\in I}Fd_i$ to them, taking into account the starting markings M_0, $\cup_{i\in I}M_{0i}$. Analysis of the behavior models of the MAS and its agents in accordance with their topological structure of static incidence relations, allows you to combine lexemes Lex, $\cup_{i\in I}Lex_i$ into a set of $SyntaS$, $\cup_{i\in I}SyntaS_i$ syntactic structures with supporting agent-based semantics of behavior, such as non-trivial paths (from two and more lexemes), trees, hammocks, loops. Syntactic structures presuppose well-defined subject and goal interpretations of their own.

For example, for subject interpretation in MAS intelligence, the formation of a non-trivial path, a chain of production steps, is based on the above formation of lexemes

$$((A_1\rightarrow^{pr,int}B_1)_{Prod},(A_2\rightarrow^{pr,int}B_2)_{Prod},\ldots,(A_n\rightarrow^{pr,int}B_n)_{Prod})\cong$$
$$(\sigma_i(A_1)^{pr,int}=B_1),(\sigma_i(A_2)^{pr,int}=B_2),\ldots(\sigma_i(A_n)^{pr,int}=B_n))$$

(here $B_1\cap A_2\neq\varnothing$ &…& $B_{n-1}\cap A_n\neq\varnothing$) or a parallel union with a common initial vertex based on its identification in an $n_{fr}+1$-slot frame

$$(Data_{sl1},Data_{sl2},\ldots,Data_{slnfr},Func()_{slfr}^{Pr,In})_{fr}\cong((Data_{sl1},Data_{sl2},\ldots,Data_{slnfr},\sigma_i^{pr,int}))_{fr}.$$

For target interpretation in the analysis of the behavior of the MAS and its agents - the presence of external inputs and outputs for the production chain

$$\rightarrow^{In}(((A_1\rightarrow^{pr,int}B_1)_{Prod},(A_2\rightarrow^{pr,int}B_2)_{Prod},\ldots,(A_n\rightarrow^{pr,int}B_n)_{Prod}))^{Out}\rightarrow\cong$$
$$\rightarrow^{In}(\sigma_i^{pr,int}(A)=B)^{Out}\rightarrow,$$

(here $\rightarrow^{In}$, ${}^{Out}\rightarrow$ is the external inference to and from the production, respectively) or the internal feasibility of frames

$$Func_{slfr}^{Pr,In}(Data_{sl1},Data_{sl2},\ldots,Data_{slnfr})_{fr}\neq\varnothing\cong\sigma_i^{pr,int}(Data_{sl1},Data_{sl2},\ldots,Data_{slnfr})_{fr}\neq\varnothing$$

(the ability to exit from it).

In the representation of the behavior of Petri nets $S(f)$, $\cup_{i\in I}S(f)_i$, the set of static syntactic structures $SyntaS^s$, $\cup_{i\in I}SyntaS_i^s$ is a set of sequences of pairs of events for subsets of positions and actions for the transitions, incident to them

$$SyntaS^s\subset W=((B(Ev)\times Pr\times Int)\times(Ac\times Pr\times Int))^*,$$
$$SyntaS_i^s\subset W_i=((B(Ev_i)\times Pr_i\times Int_i)\times(Ac_i\times Pr_i\times Int_i))^*,$$

determined based on the static relation incidence Fs, Fs_i, that is, without analyzing the markup and states of expectation, readiness and activity. Static syntactic structures $SyntaS^s$, $\cup_{i\in I}SyntaS_i^s$ are considered in the subject interpretation: a) as production paths of the form

$$(\sigma_1^{pr1,int1}(((\cup_{i11\in I11}(ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a})))=((\cup_{i11'\in I11'}(ev_{i11'}),pr_{11'e},int_{11'e}),$$
$$(ac_{11'},pr_{11'a},int_{11'a}),(\cup_{i12\in I12}(ev_{i12}),pr_{12e},int_{2e})),$$
$$(\sigma_2^{pr2,int2}(((\cup_{i21\in I21}(ev_{i21}),pr_{21e},int_{21e}),(ac_{21},pr_{21a},int_{21a})))=((\cup_{i21'\in I21'}(ev_{i21'}),pr_{21'e},int_{21'e}),$$
$$(ac_{21'},pr_{21'a},int_{21'a}),(\cup_{i22\in I22}(ev_{i22}),pr_{22e},int_{22e}))),\ldots,$$
$$(\sigma_n^{prn,intn}(((\cup_{in1\in In1}(ev_{in1}),pr_{n1e},int_{n1e}),(ac_{n1},pr_{n1a},int_{n1a})))=((\cup_{in1'\in In1'}(ev_{in1'}),pr_{n1'e},int_{n1'e}),$$

$$(ac_{n1'},pr_{n1'a},int_{n1'a}),(\cup_{in2\in In2}(ev_{in2}),pr_{n2e},int_{n2e})))) \cong$$
$$(Fs_1^{pr1,int1}(((\cup_{i11\in I11}(p_{i11},ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a})))=((\cup_{i11'\in I11'}(p_{i11'},ev_{i11'}),$$
$$pr_{11'e},int_{11'e}),(ac_{11'},pr_{11'a},int_{11'a}),(\cup_{i12\in I12}(p_{i12},ev_{i12}),pr_{12e},int_{2e})),$$
$$(Fs_2^{pr2,int2}(((\cup_{i21\in I21}(p_{i21},ev_{i21}),pr_{21e},int_{21e}),(ac_{21},pr_{21a},int_{21a})))=((\cup_{i21'\in I21'}(p_{i21'},ev_{i21'}),$$
$$pr_{21'e},int_{21'e}),(ac_{21'},pr_{21'a},int_{21'a}),(\cup_{i22\in I22}(p_{i22},ev_{i22}),pr_{22e},int_{22e}))),\ldots,$$
$$(Fs_n^{prn,intn}(((\cup_{in1\in In1}(p_{in1},ev_{in1}),pr_{n1e},int_{n1e}),(ac_{n1},pr_{n1a},int_{n1a})))=((\cup_{in1'\in In1'}(p_{in1'},ev_{in1'}),$$
$$pr_{n1'e},int_{n1'e}),(ac_{n1'},pr_{n1'a},int_{n1'a}),(\cup_{in2\in In2}(p_{in2},ev_{in2}),pr_{n2e},int_{n2e})))),$$

where

$$Fs_1^{pr1,int1},Fs_2^{pr2,int2},\ldots,Fs_n^{prn,intn}\subseteq Fs\ \&$$
$$\&\ (\cup_{i12\in I12}(p_{i12}))\cap(\cup_{i21\in I21}(p_{i21}))\neq\varnothing,\ldots,(\cup_{in-12\in In-12}(p_{in-12}))\cap(\cup_{in1\in In1}(p_{in1}))\neq\varnothing;$$

b) as part of the sequences they form, as data slots for frames

$$(((\cup_{i11\in I11}(ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a}))_{1fr},((\cup_{i21\in I21}(ev_{i21}),pr_{21e},int_{21e}),(ac_{21},pr_{21a},$$
$$int_{21a}))_{2fr},\ldots,((\cup_{in-11\in In-11}(ev_{in-11}),pr_{n-11e},int_{n-11e}),(ac_{n-11},pr_{n-11a},int_{n-11a}))_{n-1fr},\sigma_{nfr}^{pr,int})_{fr}\cong$$
$$\beta(((\cup_{i11\in I11}(p_{i11},ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a}))_{1fr},((\cup_{i21\in I21}(p_{i21},ev_{i21}),pr_{21e},int_{21e}),(ac_{21},$$
$$pr_{21a},int_{21a}))_{2fr},\ldots,((\cup_{in-11\in In-11}(p_{in-11},ev_{in-11}),pr_{n-11e},int_{n-11e}),(ac_{n-11},pr_{n-11a},int_{n-11a}))_{n-1fr})_{fr},$$

where β is the application of the step of the strategy $\sigma_{nfr}^{pr,int}$ of the agent as the identification of the initial support positions/transitions

$$(\cup_{i11\in I11}(p_{i11})),\ (\cup_{i21\in I21}(p_{i21})),\ldots,\ (\cup_{in-11\in In-11}(p_{in-11})),\ \text{то есть that is}$$
$$(\cup_{i11\in I11}(p_{i11}))\cap(\cup_{i21\in I21}(p_{i21}))\cap\ldots\cap(\cup_{in-11\in In-11}(p_{in-11})))\neq\varnothing.$$

Syntactic structures $SyntaS^s$, $\cup_{i\in I}SyntaS_i^s$ are also considered in the target interpretation: c) as having inputs and outputs to other pairs of events and actions

$$\rightarrow^{In}(\sigma_1^{pr1,int1}(((\cup_{i11\in I11}(ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a})))=((\cup_{i11'\in I11'}(ev_{i11'}),pr_{11'e},$$
$$int_{11'e}),(ac_{11'},pr_{11'a},int_{11'a}),(\cup_{i12\in I12}(ev_{i12}),pr_{12e},int_{2e})),$$
$$(\sigma_2^{pr2,int2}(((\cup_{i21\in I21}(ev_{i21}),pr_{21e},int_{21e}),(ac_{21},pr_{21a},int_{21a})))=((\cup_{i21'\in I21'}(ev_{i21'}),pr_{21'e},$$
$$int_{21'e}),(ac_{21'},pr_{21'a},int_{21'a}),(\cup_{i22\in I22}(ev_{i22}),pr_{22e},int_{22e}))),\ldots,$$
$$(\sigma_n^{prn,intn}(((\cup_{in1\in In1}(ev_{in1}),pr_{n1e},int_{n1e}),(ac_{n1},pr_{n1a},int_{n1a})))=((\cup_{in1'\in In1'}(ev_{in1'}),pr_{n1'e},$$
$$int_{n1'e}),(ac_{n1'},pr_{n1'a},int_{n1'a}),(\cup_{in2\in In2}(ev_{in2}),pr_{n2e},int_{n2e}))))^{Out}\rightarrow\cong$$
$$\rightarrow^{In}(Fs_1^{pr1,int1}(((\cup_{i11\in I11}(p_{i11},ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a})))=((\cup_{i11'\in I11'}(p_{i11'},ev_{i11'}),$$
$$pr_{11'e},int_{11'e}),(ac_{11'},pr_{11'a},int_{11'a}),(\cup_{i12\in I12}(p_{i12},ev_{i12}),pr_{12e},int_{2e})),$$
$$(Fs_2^{pr2,int2}(((\cup_{i21\in I21}(p_{i21},ev_{i21}),pr_{21e},int_{21e}),(ac_{21},pr_{21a},int_{21a})))=((\cup_{i21'\in I21'}(p_{i21'},ev_{i21'}),$$
$$pr_{21'e},int_{21'e}),(ac_{21'},pr_{21'a},int_{21'a}),(\cup_{i22\in I22}(p_{i22},ev_{i22}),pr_{22e},int_{22e}))),\ldots,$$
$$(Fs_n^{prn,intn}(((\cup_{in1\in In1}(p_{in1},ev_{in1}),pr_{n1e},int_{n1e}),(ac_{n1},pr_{n1a},int_{n1a})))=((\cup_{in1'\in In1'}(p_{in1'},ev_{in1'}),$$
$$pr_{n1'e},int_{n1'e}),(ac_{n1'},pr_{n1'a},int_{n1'a}),(\cup_{in2\in In2}(p_{in2},ev_{in2}),pr_{n2e},int_{n2e}))))^{Out}\rightarrow;$$

d) as possessing the property of liveliness for a set of frame slots with pairs of events and actions

$$\sigma_{nfr}^{pr,int}(((\cup_{i11\in I11}(ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a}))_{1fr},((\cup_{i21\in I21}(ev_{i21}),pr_{21e},int_{21e}),(ac_{21},$$
$$pr_{21a},int_{21a}))_{2fr},\ldots,((\cup_{in-11\in In-11}(ev_{in-11}),pr_{n-11e},int_{n-11e}),(ac_{n-11},pr_{n-11a},int_{n-11a}))_{n-1fr})_{fr}\neq\varnothing\cong$$
$$Fd_{nfr}^{pr,int}(((\cup_{i11\in I11}(p_{i11},ev_{i11}),pr_{11e},int_{11e}),(ac_{11},pr_{1a},int_{1a}))_{1fr},((\cup_{i21\in I21}(p_{i21},ev_{i21}),pr_{21e},$$
$$int_{21e}),(ac_{21},pr_{21a},int_{21a}))_{2fr},\ldots,((\cup_{in-11\in In-11}(p_{in-11},ev_{in-11}),pr_{n-11e},int_{n-11e}),(ac_{n-11},pr_{n-11a},$$
$$int_{n-11a}))_{n-1fr})_{fr}\neq\varnothing.$$

When verifying the MAC behavior model, syntactic structures $SyntaS$, $\cup_{i\in I}SyntaS_i$ take a special form of identifiers - external behavioral characteristic neighborhoods for: a) reference states Id, $\cup_{i\in I}Id_i$; b) unobservable (internal) fragments of behavior, both realized by $Real$, $\cup_{i\in I}Real_i$ from external control points, and transported by $Trans$, $\cup_{i\in I}Trans_i$ to them; c) inherited Inh, $\cup_{i\in I}Inh_i$, mutually mapped between adjacent levels of pairs of fragments of behavior, preserving the properties of characteristic.

Another special kind of syntactic structures for verification are connectors $Conn$, $\cup_{i\in I}Conn_i$ - behavioral fragments of mutual reachability and direct inheritance, shortest paths for syntactic structures and support states in composition, interlevel encapsulation containers and detail structures.

Preparation of verification of the behavior of Petri nets $S(f)$, $\cup_{i\in I}S(f)_i$ saves a set of static verified syntactic structures $SyntaS^s$, $\cup_{i\in I}SyntaS_i^s$ - identifiers of the above types of Id^s, $\cup_{i\in I}Id_i^s$, $Real^s$, $\cup_{i\in I}Real_i^s$, $Trans^s$, $\cup_{i\in I}Trans_i^s$, Inh^s, $\cup_{i\in I}Inh_i^s$ and connectors $Conn^s$, $\cup_{i\in I}Conn_i^s$ with the peculiarity of using the concept of support positions / transitions instead of the concept of support states.

Dynamic verifiable syntactic structures $SyntaS^d$, $\cup_{i\in I}SyntaS_i^d$ are formed from static syntactic structures, as their narrowing based on the application of dynamic incidence relations Fd, $\cup_{i\in I}Fd_i$ to them, taking into account dynamic lexemes Lex^d, $\cup_{i\in I}LexS_i^d$ and starting markings M_0, $\cup_{i\in I}M_{0i}$.

The subsequent analysis of MAS behavior integrates subsets of the previously defined syntactic structures $SyntaS$, $\cup_{i\in I}SyntaS_i$ into fragments of MAS behavior - subject semantic behavior networks SeN, $\cup_{i\in I}SeN$, which is performed on the basis of relations-links between these structures, in particular, based on identity, adjacency, incidence, reachability, inheritance, and the *Conn* connectors they form.

The semantics of integration suggests subject and target interpretation for the behavior of the MAS, which scales the interpretation of syntactic structures and emphasizes complex non-linear inference.

So for the subject interpretation of the intelligence of the MAS in the semantic networks SeN, $\cup_{i\in I}SeN_i$, a multi-step, network multi-level logical inference is performed based on the previously indicated relations-links for the elements of this semantic network - productions and frames, and for the target interpretation in the analysis of the behavior of the MAS - the determination at logical inference in the semantic networks SeN, $\cup_{i\in I}SeN_i$ of the properties of external, direct and reverse reachability and heritability of productions or frames, liveliness and safety of MAC behavior.

When preparing the verification of the behavior of MAC based on Petri nets $S(f)$, $\cup_{i\in I}S(f)_i$ static semantic networks SeN^s, $\cup_{i\in I}SeN_i^s$ are also formed on the basis of the corresponding syntactic structures $SyntaS^s$, $\cup_{i\in I}SyntaS_i^s$. Dynamic semantic networks SeN^d, $\cup_{i\in I}SeN_i^d$ are formed from static ones, as their narrowing based on the application of dynamic incidence relations Fd, $\cup_{i\in I}Fd_i$ to them, taking into account the dynamic syntactic structures $SyntaS^d$, $\cup_{i\in I}SyntaS_i^d$ and starting markings M_0, $\cup_{i\in I}M_{0i}$.

When verifying MAS behavior, including verification of knowledge models knS and knS_i and behavior models in the form of Petri nets $S(f)$, $\cup_{i\in I}S(f)_i$, such fragments of behavior are fragments of subject dynamic semantic networks SeN^d, $\cup_{i\in I}SeN_i^d$ - generated by verifiers, in particular, they can be verification primitives – identifiers of the specified types Id^s, $\cup_{i\in I}Id_i^s$, $Real^s$, $\cup_{i\in I}Real_i^s$, $Trans^s$, $\cup_{i\in I}Trans_i^s$, Inh^s, $\cup_{i\in I}Inh_i^s$ and connectors $Conn^s$, $\cup_{i\in I}Conn_i^s$, as well as networks and hierarchies of verification fragments of behavior – corresponding nS and SI models (2), which are networked and end-to-end displayed by sequences of verification fragments in networks of levels and between levels.

The dynamic logical inference in such semantic networks SeN^d, $\cup_{i\in I}SeN_i^d$ is formed on the basis of the selection and formation of transitions in accordance with the dynamically weighted previously presented relations-links based on Fd, $\cup_{i\in I}Fd_i$ for fragments of behavior, in particular, relations of identification, adjacency and incidence, reachability, inheritance of the components of these fragments.

The dynamic weights of the components of the reconfigurable Petri nets $S(f)$, $\cup_{i\in I}S(f)_i$, - positions, transitions, relations-links, markings suggest intelligent statistical and priority analysis, search for new options with redefinition of choice weights. The possibility of defining a knowledge model as a subject semantic network and logical inference in it is formed by using extended hierarchical Petri nets (1), weighted by adaptive statistical probabilities of experience and priorities dynamically changing from zero (no connection) to one (there is only one connection), temporal intervals.

3.4. Verification Model of Intelligence

The complete model of behavior of MAS contains all of its subject properties and mechanisms, including intellectuality. The extended hierarchical Petri net, used as a model of the behavior of MAS, allows spatial and temporal decomposition. As noted earlier, many problems of the behavioral analysis of discrete systems, including the verification of DIS and MAS, are *NP*-complex, and the decomposition of the MAS model, in particular, the functional decomposition, that highlights certain properties and mechanisms of behavior and complements the spatial and temporal ones, can also significantly reduce this complexity with a slight decrease in the completeness of analysis and verification.

In this work the properties and mechanisms of the intelligence of MAS are selected for verification. In this case, the original MAS model, in particular, in the form of an extended hierarchical Petri net, can be reduced to select intelligence on the assumption, that the properties and mechanisms not directly related to it are reliable, that is, they are not subjected to analysis and verification. In this case, all fragments of the Petri net, that are not related to the properties and mechanisms of intelligence, as well as to the points of control and observation of their behavior, are subjected to intelligent reduction.

As shown above, the dynamic extended hierarchical Petri nets, representing intelligence in the behavior of MAS, make it possible to determine the goals of analysis and verification, the corresponding tasks-strategies, models, methods, criteria-parameters. At the same time, such Petri nets, in addition to functional decomposition for intelligence, retain the possibility of decomposition, both spatial – in a distributed environment for the placement of agents of MAS, and temporal – in a multi-level representation of behavior processes, including the hierarchical arrangement of task-scenarios, methods, actions, functions with their conditions and events.Leaving out of consideration the process of intelligent reduction (minimization) of the Petri nets and without losing the generality of the verification models, we can accept, as a reduced, the Petri net given earlier (1).

Then, for the network and hierarchical representation of the MAS model, in turn, it is possible to accept the network and hierarchical models, based on extended Petri nets, also given earlier (2).

In this regard, a model for verifying MAS intelligence, which determines the conditions for its implementation under the assumption of check and recognition of reference behavior, based on identification, can be specified on the basis of abstract (component), network and hierarchical verification models of extended Petri nets [23, 25]:

$$cs_i=((W_i^{\wedge},Ci_i,Cp_i,Lp_i,Cf_i,\ Sg_{S(f)i}),$$
$$ncS=(CS,\ node^{\wedge},\ R_T^{-1}{}_{(node^{\wedge})},\ Tr_{T(node^{\wedge})},\ Sg_{cnS},\ R_{(S(f)^{\wedge})},\ Tr_{(S(f)^{\wedge})}),$$
$$2cIS = (cS,\ \cup_{i\in I}\ cS_i^p,\ \cup_{j\in J}\ cS_j^t,\ Sg_{icS}). \tag{5}$$

The set of cM_i methods of behavioral verification of MAS intelligence, as special, asynchronous-event parallel procedures that satisfy the conditions of the presented component cS_i, network cNS and hierarchical $2cIS$ verification models, can be formally defined using special Petri nets, the main features of which are presented in the above presented models of agents, goals, strategies, knowledge.

4. Analytical and Experimental Complexity

The overall computational complexity of the k-decomposition for models of verification of intelligence on base of hierarchical Petri nets is reduced exponentially by a factor of k, is showed as Figure 2 and determined by the upper bound:

$$c \le k(((lm)^{n/k}2^{(n-1)/k})(2n/k+3)+n/k(2m+1)+m(n/k-1)(2n/k+l))/log2(2nm/k), \tag{6}$$

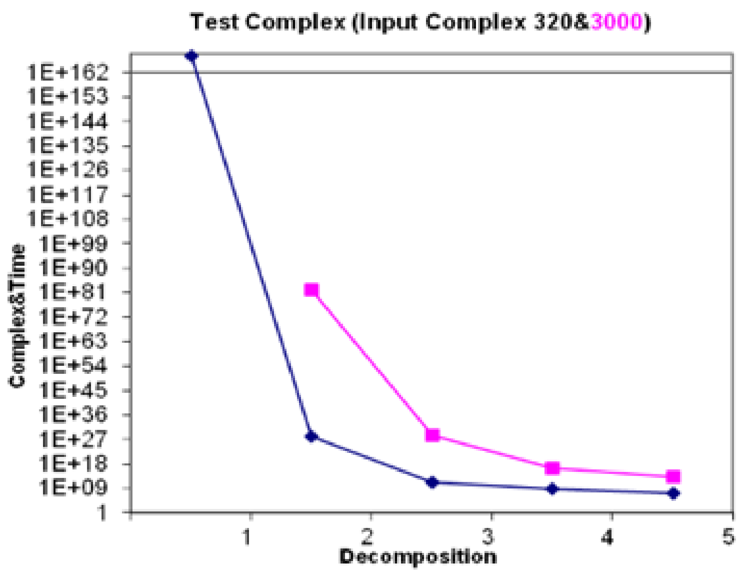

Figure 2: Decomposition dependence for upper computational complexity of verification

The experimental test verification was held for real MAC of monitoring, control and video surveillance showed (table 1).

Table 1.
Experimental complexity of verification on the degree of k-decomposition

Intelligent MAS Systems	Decomposition Degree k	Input reduced complexity 50	Input reduced complexity 500
1 MASMC Module	8	14296	1,43E+17
2 MASVS Module	10	1825	1,09E+11

The comparison of test verification, based on Petri net, for dynamic intelligence of MAS of monitoring and control (MASMC) and video surveillance (MASVS) confirmed decrease in computational complexity of check.

5. Conclusion

The choice of properties and actions for behavior verifying the intelligence of MAS and its agents illustrates the possibility of a decrease in the computational complexity of the check analysis of the verification model exponentially by a factor of *k* and the subsequent reduction of the verification time in a certain set of verification methods of intelligence, corresponding to this model due to functional *k*-decomposition. When using software and hardware simulation tools of the Intel i5-7400T level, RAM 8.00 GB, the decomposition verification time achieved for real MAC components was determined in the range of 2-15 milliseconds, for MAC as a whole - 0.5-25 hours. A significant reduction in time in comparison with the upper estimates was achieved by taking into account the special structural and functional properties of real MAS. However, the decomposition does not remove the *NP*-complexity of the behavioral verification generally represented by these upper bounds. In the future, in order to further reduce this complexity, it seems appropriate to develop behavioral verification of MAS intelligence through the use of fuzzy knowledge models and logical inference.

6. References

[1] P.L. Dordal, An Introduction to Computer Networks. Release 2.0.5. Department of Computer Science Loyola University Chicago, 2021, 947 p.
[2] S. Pakari, E. Kheirkhah, et, al., Web-service Discovery Methods and Techniques: a Review. International Journal of Computer Science, Engineering & Information Technology, 2014, (4) pp. 1–14.
[3] Manoj Kumar Mishra1, Yashwant Singh Patel, Moumita Ghosh, G. B. Mund, A Review and Classification of Grid Computing Systems. International Journal of Computational Intelligence Research, Volume 13, Number 3 2017, 13(3). pp. 369-402.
[4] Xie Jing, Liu Chen-Ching, Multi-agent systems and their applications. Journal of International Council on electrical engineering, 2017, 7(1). pp. 188–197.
[5] B.M.K. Prasad, K. Singh, S.S. Pandey, R. O'Kennedy, Communication and Computing Systems (1st ed.). CRC Press, 2019, 644 p.
[6] G. A.Tsihrintzis, M.Virvou, C. Jain. Lakhmi, Intelligent Computing Systems: Emerging Application Areas. (1st. ed.). Springer Publishing Company, Incorporated, 2016, 368 p.
[7] N. Skehin, Chung Jaewoo, Mobile Computing Systems and Applications. IEEE Pervasive Computing, 2011, 10. pp. 80-83.
[8] C. Borcea, Iyer Deepa, PL Kang, Saxena Akhilesh, Iftode Liviu, Cooperative Computing for Distributed Embedded Systems. In proc. of International Conference on Distributed Computing Systems, 2002. pp. 227-236.

[9] T. Hovorushchenko, Methodology of Evaluating the Sufficiency of Information for Software Quality Assessment According to ISO 25010. Journal of Information and Organizational Sciences, 2018, 42(1). pp. 63-85.

[10] Kumar Jitendra, Shinde Vikas, Kalra Mukta, Availability and Reliability Analysis of Computer Systems. International Journal of Control Theory and Applications, 2019, 10. pp. 266-275.

[11] Joachim Hänsel, Holger Giese, Towards collective online and offline testing for dynamic software product line systems. In proc. of the 2nd International Workshop on Variability and Complexity in Software Design (VACE '17). IEEE Press, 2017. pp. 9–12.

[12] J.-O. Adetokunbo, C. Ojieabu, Automated Fault Detection and Identification System for Computer Networks. 2017.

[13] L. Gomes, J.M. Fernandes, Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation. 2010, 494 p.

[14] A. Drozd, M. Drozd, O. Martynyuk, M. Kuznietsov, Improving of a Circuit Checkability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems. CEUR Workshop Proceedings, 2017, 1844. pp. 654–661.

[15] R. M. Hierons, K. Bogdanov, J. P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Luttgen, A. J. H. Simons, S. Vilkomir, M. R. Woodward, H. Zedan, Using formal specifications to support testing. ACM Comput. Surv., 2009, 41. pp. 9:1-9:76.

[16] A.P. Wigderson, NP and mathematics – a computational complexity perspective. Retrieved 3 Apr. 2021.

[17] K. Lam. William, Hardware design verification: simulation and formal method-based approaches. Prentice Hall, 2005, 624p.

[18] M. Suma, K. Gurumurthy, Fault Modeling of Combinational and Sequential Circuits at Register Transfer Level. International Journal of VLSI Design & Communication Systems. 2011, 2.

[19] D. Faitelson, R. Heinrich, S. Tyszberowicz, Functional Decomposition for Software Architecture Evolution. In Book: Model-Driven Engineering and Software Development, 2018. pp. 377-400.

[20] R. Harrison, D. Vera, B. Ahmad, Engineering Methods and Tools for Cyber-Physical Automation Systems. In proc. IEEE 104, 5, May, 2016. pp. 973-985.

[21] D. Weyns, F. Michel, V. Parunak, M. Schumacher, A. Ricci, A. Brandao, C. Carrascosa, O. Dikenelli, S. Galland, A. Pijoan, J. Saunier, V. Urovi, F. Zambonelli, Agent Environments for Multi-Agent Systems – A Research Roadmap. Linnaeus University Sweden, KU Leuven Belgium. 2015. pp. 1-19.

[22] B. Aizier, S. Lizy-Destrez, C. Seidner, V. Chapurlat, D. Prun, J. L. Wippler, xFFBD: towards a formal yet simple and complete functional modeling technique for system designers. In proc. INCOSE 2012, 22nd Annual International Council on Systems Engineering Symposium, Rome, Italy, 2012. pp. 170-183.

[23] O. Martynyuk, O. Drozd, Tamim Ahmesh, Bui Van Thuong, A. Sachenko, H. Mykhailova, M. Dombrovskyi, Hierachical Model of Behavior On-line Testing for Distributed Information Systems. The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'19), Metz, France, 2019, 2. pp. 724-729.

[24] D.Weyns, Architecture-Based Design of Multi-Agent Systems. Springer. 2010, 224 p.

[25] O. Martynyuk, O. Drozd, H. Stepova, Thuong Van Bui, D. Martynyuk, L. Sugak, Model of Multiagent Cooperation for Behavioral Testing. In proc. 2021 IEEE East-West Design & Test Symposium (EWDTS) Batumi, Georgia, 2021. pp. 224-228.