 doi:10.20944/preprints202101.0028.v1

Article

Model of problem-solving for robotic actions planning

Oleksandr Tsymbal [1], Paolo Mercorelli [2,*]

[1] Kharkiv National University of Radio Electronics, Kharkiv, Ukraine; oleksandr.tsymbal@nure.ua
[2] Leuphana University, Lunenburg, Germany; paolo.mercorelli@leuphana.de
* Correspondence: paolo.mercorelli@leuphana.de; Tel.: +49.4131.677-1896

Abstract: Proposed article deals with analyses on trends of problem-solving robotic applications in manufacturing, especially in transportations and manipulations. Intelligent agent-based manufacturing systems with robotic agents are observed. Intelligent core of such units must be able to propose the plan of problem-solving in form of strategy. The logical model of adaptive strategies planning for intelligent robotic system is described in form of predicates with presentation of data processing on base of set theory descriptions. Dynamic structure of workspace and possible change of goals are considered as reasons for functional strategies adaptation. Proposed formal descriptions are supported by model of mobile robotic platform, acting in warehouse.

Keywords: adaptation; problem-solving; robotics; predicates; manufacturing system

1. Introduction

This paper is an extension of work originally presented in SPEEDAM 2020 [1].

Modern manufacturing systems are described by intensive application of information technologies on base of computer networks, artificial intelligence and digital technologies and must correspond to requirements of mobility, of fast response to changing quality of product, of small sizes, specific, individual, customer and environmental demands. Last two decades industrial engineers and scientists spend a lot to research the advanced production systems and their influence to global market [6].

The part of mentioned concepts of manufacturing systems was widely applied in practice and supplied the essential decrease of design development time. Others – are at the research and conceptualization stage, but, in any case, their base is in the designer (also manager and manufacturer) intelligence, that creates the manufacturing intelligence.

The current state of FIS can be described by participation of numerous units of equipment, including technological, transportation and warehouse units, also humans, acting in workspaces. Therefore, such systems can be described and simulated as multi-agent, on their origins and functioning. It is completely corresponds to modern manufacturing concepts.

The open architecture of multi-agent systems (MAS) becomes the basic direction of distributed artificial intelligence development. From practical point of view these agents are considered as self-controlled software objects with self-estimation system and independent problem-solving tools for own need and for other agents (by requests) [2].

The conceptual agent model consists of 4 components:

- perception, which is a source for information about external world;
- effector, which is an agent's interface to change world or have effect in agent's community; communication system, which is the tool for information exchange between other members of agent's community;
- aims, which are the list of the goals for agent's execution.

Any manufacturing system can be considered as a sort of MAS, because it consists of various kinds of equipment (with/without intelligent features), possibly including manipulation and transportation robots, human personal and united by automated control system. Modern manufacturing systems (which are mainly decentralized) are typical

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

application of MAS. There are a lot of benefits of such application: distributed processing of information in a way unlike the single big systems; there is supplied the quality increase for manufacturing systems by learning and interaction of objects; there is supplied the method of plant integration.

For condition of manufacturing the agent is an object with certain intellectual level, which can be as physical (worker, machine) as logical object (task, directive, order). In this way, robot with manipulation and transportation functions looks as just ideal manufacturing agent. In [3] we can see presentation of conceptual model of such agent.

Figure 1 presents the model of MA. It includes the library of procedures (the experience of system on form actions). inference drive (or problem-solver), knowledge base (more general than procedures), perception processor with possibility to communicate with sensors of robots. Simulation component supplies estimation of possible results for the activity of MA. Like any computing systems MA includes memory and communication units. Coordination subsystem checks the internal functions of MA, receives the queries for coordination from other MA [3].

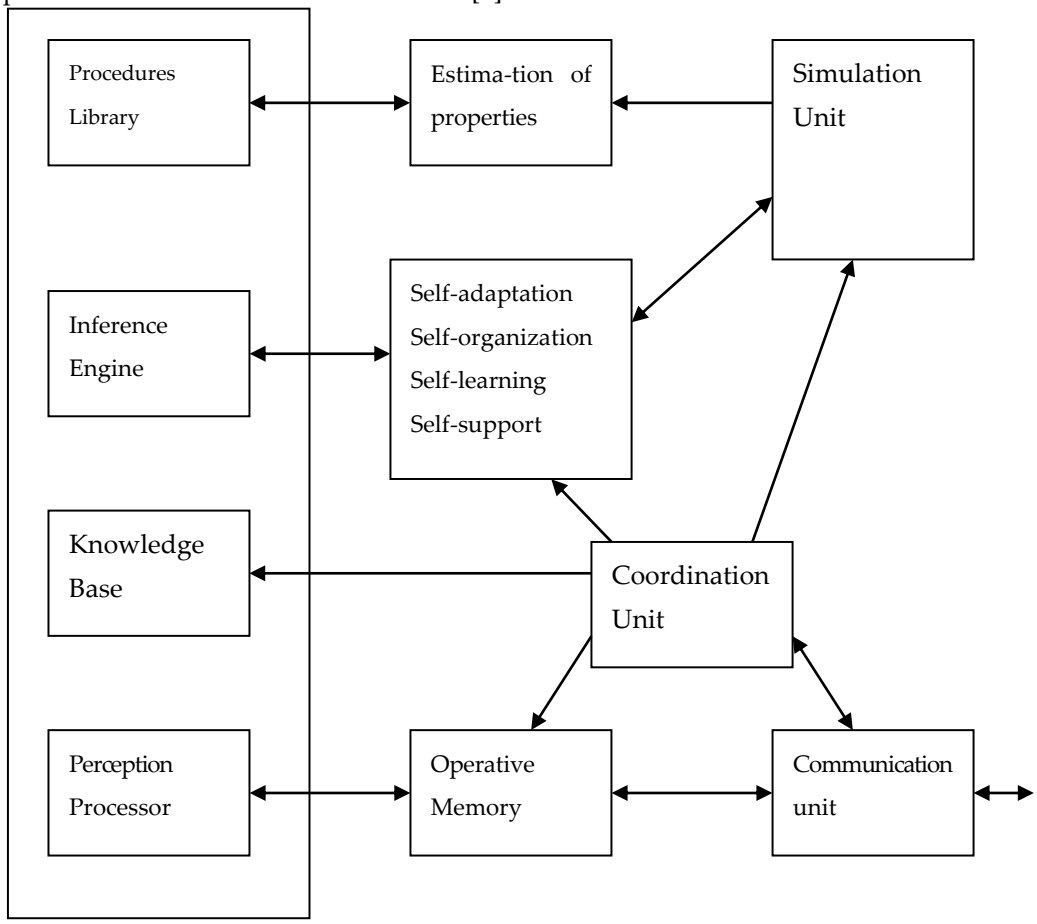

Figure 1. Manufacturing Agent conceptual model [3]

There are three commonly known types of MA-systems; functional, heter-hierarchy-based, workspace-based architecture. Heter-hierarchy-based multi-agent architecture has greater perspectives because of its self-organization, scaling, error-stability, decreased complexity, greater flexibility, and decreased costs, parallelism, Internet-compatibility, possibility of virtual manufacturing forming. In hetero-hierarchy multi-agent manufacturing system different MA's interact at level of network nodes without fixed relationships of chief-worker type. The system's aims are achieved by interaction of MA's. According to general structure of manufacturing [5] proposes the architecture of manufacturing system on agent-based concept.

From multi-agent manufacturing systems implementation point of view, the interaction of agents at all the manufacturing levels isn't the only key point. Also important is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

implementation of all the agents at all the levels. Very important task are resource planning, technological processes design, technological processes schedules development. In such conditions MA can be implemented in virtual (as an automated control system function set) or physical (as industrial or transport robot) agents, able to analyze the constructive and technological specifications of manufacturing for specific workplace (WS) in FMS, to monitor the production process, to check-up the manufacturing technology acceptance, to respond to predicted or unpredicted manufacturing situations, to supply the selected functions of operative manufacturing control.

In terms of control strategy, recent publications in the field of robot control for manufacturing indicated progress and interesting results, see [11] and [12]. The paper is organized in the following way. Section 2 considers the proposed adaptive strategies planning. The conclusions closes the paper.

2. Adaptive strategies planning for intellectual robotic systems

From description of manufacturing agent conception, we can see need to describe creation and activity of problem-solving component. Such component must include the set of operative procedures (actions) with common knowledge support; inference engine as core of problem-solving; dynamic database with information on surrounding workspace of manufacturing system.

Actions of manufacturing robotic agent can be described as transformation of states of robotic platform and external workspace (WS).

Robotic agent (RA) can be described by number of sets X, D, S of platform states, of robotic system decisions and of WS.

Correspondently, $x_i \in X, d_i \in D, s_i \in S$ can be introduced as atoms for model, which describes robotic agents and it's WS. For description of model we can also introduce standard operations $\neg,\wedge,\vee,\rightarrow,\leftrightarrow$ and Well-Formed Formulas on base of them:

$$\neg x, x \wedge y, x \vee y, x \rightarrow y, x \leftrightarrow y.$$

To describe the theory sets $X, D, S$the functions and predicated are introduced.

Transition of states for RA: $x_i = f(x_0, \ldots, x_{i-1})$,

Transition of states for WS: $s_i = f(s_0, \ldots, s_{i-1})$.

To define the logics of problem-solving system the set of predicates is introduced:

$$pt(x_i), pt(s_i), pt(d_i),$$

$$pt(x_i, s_i), pt(x_i, d_i), pt(d_i, s_i), pt(x_i, d_i, s_i)$$

These predicates define:

$pr(x_i, s_i) \subset pt$ – states of RA in workspace WS,

$ps(x_i, s_i) \subset pt$ – states of WS in system of RA,

$pa(x_i, s_i) \subset pt$ – actions of RA inside WS,

$pg(pr, ps) \subset pt$ – goals of RA inside WS.

Every goal of RA is formulated as a new (or existing) state of RA or WS:

$$pg(pr, ps) \leftarrow (pr(x_i, s_i) \vee ps(x_i, s_i)).$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

Database of RA is combines: $pr(x_i), pr(x_i, s_i), ps(s_i), ps(x_i, s_i)$.

Knowledge base of RA includes possible actions $pa(x_i, s_i)$ of RA in WS.

Predicate $pa(x_i, s_i)$ is a strategy, which solves goal $pg(pr(x_i, s_i), ps(x_i, s_i))$, if there is exist such conjunction of RA actions $pa(x_i, s_i)$, which supplies $pg(pr(x_i, s_i), ps(x_i, s_i))$:

$pg(pr(x_i, s_i), ps(x_i, s_i)) \leftarrow pa^0(x_i, s_i) \wedge pa^1(x_i, s_i) \wedge .. \wedge pa^{n-1}(x_i, s_i)$,
or

$pg(pr(x_i, s_i), ps(x_i, s_i)) \leftarrow \wedge_{i=0}^{n-1} pa^i(x_i, s_i)$,
and besides:

$\exists f, f \in F: x_i = f_i(x_{i-1}, s_{i-1}), \exists \psi, \psi \in \Psi: x_i = \psi_i(x_{i-1}, s_{i-1})$.

Therefore, $pa(x_i, s_i) = T\|f_i + \psi_i\|$.

The problem-solving process is sequence of *m*-alternatives to reach goals of RA:

$pg^0(pr, ps) \leftarrow pg_0^0(pr_0, ps_0, pa_0) \wedge pg_1^0(pr_1, ps_1, pa_1) \wedge ..$

$\wedge pg_{n-1}^0(pr_{n-1}, ps_{n-1}, pa_{n-1}) = \wedge_{i=0}^{n-1} pg_i^0(pr_i, ps_i, pa_i)$

……

$pg^m(pr, ps) \leftarrow pg_0^m(pr_0, ps_0, pa_0) \wedge pg_1^m(pr_1, ps_1, pa_1) \wedge ..$

$\wedge pg_{n-1}^m(pr_{n-1}, ps_{n-1}, pa_{n-1}) = \wedge_{i=0}^{n-1} pg_i^m(pr_i, ps_i, pa_i)$.

As a result global (final) goal is defined as follows:

$pg^{total}(pr, ps) \leftarrow \vee_{j=0}^{m-1} \wedge_{i=0}^{n-1} pg_i^m(pr_i, ps_i, pa_i)$.

Every RA starts planning with development of initial plan of actions in WS. It includes the next transitions:

$pr(x_1, s_1) \leftarrow pa_0^0(pr(x_0, s_0) \vee ps(x_0, s_0)), pr(x_2, s_2) \leftarrow pa_1^0(pr(x_1, s_1) \vee ps(x_1, s_1))$,
……..

$pr(x_n = Y, s_n) \leftarrow pa_{n-1}^0(pr(x_{n-1}, s_{n-1}) \vee ps(x_{n-1}, s_{n-1}))$.

Such transitions can be correct for static WS, but becomes practically wrong if WS is dynamic with impossibility to reach desired state:

$pr(x_i, s_i) \neq pa_i^0(pr(x_{i-1}, s_{i-1}) \vee ps(x_{i-1}, s_{i-1}))$.

In this case strategy must be modified:

$pr(x_i, s_i) \leftarrow pa_i^*(pr(x_{i-1}, s_{i-1}) \vee ps(x_{i-1}, s_{i-1}))$,

$pr(x_{i+1}, s_{i+1}) \leftarrow pa_{i+1}^*(pr(x_i, s_i) \vee ps(x_i, s_i))$,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

with general result:

$$pg^{m*}(pr,ps) \leftarrow pg_0^{m*}(pr_0,ps_0,pa_0) \wedge pg_1^{m*}(pr_1,ps_1,pa_1) \wedge \ldots$$

$$\wedge pg_{n-1}^{m*}(pr_{n-1},ps_{n-1},pa_{n-1}) = \wedge_{i=0}^{n-1} pg_i^{m*}(pr_i,ps_i,pa_i).$$

Let's define RA states as set $X = \{X^0, X^1, \ldots, X^{n-1}\}$.

Let the automated control system of RA in the process of execution of the decision provides the transformation of the initial state$state(x_0^0, x_1^0, x_2^0, \ldots, x_{n-1}^0)$ into a specific goal state $state(x_0^m, x_1^m, x_2^m, \ldots, x_{n-1}^m)$.

If the system (RA and the WS around it) at the initial time is a set of arguments $x_0^0, \ldots, x_n^0$ and is characterized by a state $state(x_0^0, x_1^0, x_2^0, \ldots, x_{n-1}^0)$, then considering the discrete process of planning strategies, which consists of individual actions $action_0, \ldots, action_k$, we can indicate that the transition from one discrete state to another is a certain relationship between objects:

$$state(x_0^1, x_1^1, \ldots, x_n^1) \leftarrow action_0(state(x_0^0, x_1^0, \ldots, x_n^0)),$$

here *state* - a relationship (predicate) that characterizes the state of the system as a whole, *action (state)* - which means the action of transition from one state to another.

All activities related to the transition from one state of the system to another (by executing a list of solutions) are a set of predicates:

$$state(X^1) \leftarrow action_0(state(X^0)),$$

$$state(X^2) \leftarrow action_1(state(X^1)),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$state(X^{n-1} = Y) \leftarrow action_{n-2}(state(X^{n-2})).$$

Thus, the goal of a strategy planning system is to find the appropriate number of $action_i$ that would satisfy the $state_i$ conditions of the system. The choice of the $action_i$ action to convert the $state(X^i)$ to state $state(X^{i+1})$ is made to ensure compatibility of the *action* arguments and the corresponding state X^{i+1}, which in practice will mean the possibility of implementing the state (local goal) X^{i+1} by performing of *action*:

$$X^{p+1} \leftarrow action\ (X^p),$$

with X^{p+1} – as possible result of action *action* for condition of state X^p.

The predicate scheme is adaptive if the components of the antecedent (right part of the predicate expression) and the result of the scheme (consequent) change depending on changes in the state of the ASC and the working space (RP):

$$state(Y) \leftarrow action\ (S_0), action\ (S_1), \ldots, action\ (S_{n-1}),$$

here Y – final goal of RA; S_0, S_1, … S_{n-1} – the sequential states of control system.

However, in case of dynamics of WS (world of robot) the state of robot can also be transformed (such changes aren't obligate, but probable) and there is possible situation, when for some state $state(X^{i-1})$ action $action_{i-1}(state(X^{i-1}))$ will not transform system to state $state(X^i)$, thus:

$$state(X^i) \neq action_{i-1}(state(X^{i-1}))$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

The first look to these conditions is to find such predicate *action*, which satisfy the condition of expression. However, the actual number of real actions is limited (unlike the number of states), and the solution is in search of such discrete sequence $\overrightarrow{action}$ (vector of predicated), which satisfies to the goal of system.

Therefore, if there is set X of world's objects and X^0 is initial set states, then to supply goal state Y it's need to make plan with sequence of actions, expressed by predicates *action*, and with state of system – predicate *state*:

$$state(X^0),$$

$$state(X^1) \leftarrow action_0(state(X^0)),$$

$$state(X^2) \leftarrow action_1(state(X^1)),$$

$$\ldots\ldots\ldots\ldots$$

$$state(Y) \leftarrow action_{n-1}(state(X^{n-1})).$$

If, at some step i, the state X^i is unobtainable, that $state(X^i) \neq action_{i-1}(state(X^{i-1}))$, then the adaptive strategies planning system must generate the new order of action predicates $\overrightarrow{action}$, that will satisfy the changes of WS:

$$state(X^{i1}) \leftarrow action^1_{n-1}(state(X^{i-1})),$$

$$state(X^{i2}) \leftarrow action^2_{n-1}(state(X^{i-1})),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$state(Y) \leftarrow action^{m-1}_{n-1}(state(X^{n-1})).$$

The similar situation is when DMS has information that the goal of system is changed. It means, that the goal state *state(Y)* will be changed to some *state(Y^i)*. Here, two variants are possible:

а) the information on the change of goal comes at moment, when system is in the state i – *state(x^i)* and there is possible to generate plan $\overrightarrow{action}$ to transit from x^i to state y^i;

б) the information on the change of goal comes at moment, when system is in the state i – *state(x^i)*, but the generation of plan $\overrightarrow{action}$ is possible only from state *state(x^{i-k})*, where $k \leq i$, so, to generate plan, system must return to previous states, possibly, up to state *state(x^0)*.

Again, it will need generation of new order of predicate actions.

According to definition, the plan of decision will consist of sets {*action$_0$*, *action$_1$*, …, *action$_n$*}, and the entire plan – of all the plans, developed during the problem-solving. The adapted decision plan will be the expression:

$$plan^{adaptive}(Y) \leftarrow action_0(state(X^0), action_1(state(X^1), \ldots, action_{n-1}(state(X^{n-1}))))$$

.

Such plan in final decision of adaptive DMS.

The developed plan will be changed up to execution of last subgoal of planned order and plan's adaptation can be considered as its essential specification.

Also note on need to evaluate the actions, proposed by DMS in a sequence of predicates.

As is known, a predicate has verity value. In classics, it's a mapping of n arguments to verity value. While, the fuzzy sets theory directs to possible introduction of fuzzy

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

predicate term [9], the classic predicate has only two values *true* and *false*. At this point, the DMS transition from one state *state*(X_{i-1}) to state *state*(X_i) also has values of *true* or *false*, so system transfers to new state or not. Probably, it's difficult to predict the state of whole system even for ideal case, more to give the simple system evaluation in binary values of *true* or *false*. Rather, the verity or falsity will describe the particular system parameters. As to predicated theory, the robot's world can be described as relation set between world's objects, for instance, *is_a* – membership to the object's type, *is_at* – one object positioning near the other, *stands* – object's being in some state, etc.

The flexible integrated assembling system (FIAS) contains assembling automatic (soldering) machines, assembling-transport robots, storehouses, defining set $eq_i \in Eq,\ i = 0...n-1$. The purpose of FIAS is a execution of assembling technological process (modules) of radio-electronic devices, in particular for printed circuit board *M*.

Here $M=<B, Ch, T, R, C, L, \ldots>$, where *B* – the printed circuit board, *Ch* – microchips, *T* – semiconductor devices, *R* – resistors, *C* – capacitors, *L* – inductances.

The configuration of device is determined by its construction design M^G, which defines the purpose location of elements at printed circuit board. Actually the module (board) is a rectangular matrix, filled by elements of set *M* (shown at figure 4)

Initially, M_0 is zero matrix. FIAS generates decisions $d_k \in D,\ k = 0,\ldots, l-1$, which are implemented by actions (technological transitions): $a_k \in A,\ k = 0,\ldots, l-1$. Decision $\vec{D}$ for the order of assembling operations execution is a sequence of operations $a_i \in A,\ i = 0...l-1$, which are in the settings to board *B* of some elements from sets *Ch*, *T*, *R*, *C*, *L*…, for example:

$$\vec{D} = \{Ch_0,\ Ch_1, T_0, T_1, R_0, Ch_3, C_0, L_0, \ldots\}$$

To reach the goal state M^G there are transformations $M_i = f_i(Eq,\ D_i,\ M_{i-1})$:

$$M_0 \to M_1 \to M_2 \to \ldots \to M^G$$

The filling of *M* is defined by order of assembling operations. Such order is set by design projectM^G, technological rules Tr, abilities of technological equipment E. Therefore, $\vec{D} = g(M^G,\ Tr,\ E)$. The purpose of search is to find such sequence of transitions$f_1, \ldots, f_n$, that provides transformation from initial state M_0 to goalM^G.

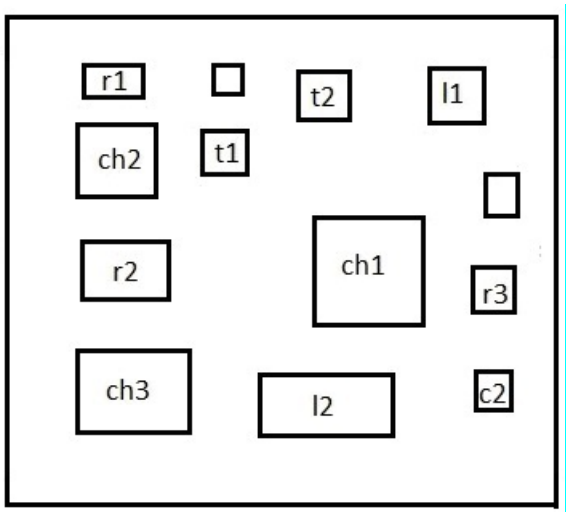

Figure 2. Workspace for manipulation task

The strategies planning process is an act of constant comparation of system's goal to the current state, current possibilities. The strategies planning process according to dis-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

continuity of operations is also must be discrete, must correspond to goal's achievement and implement the particular technological operations.

In general, the strategies planning process is a mapping of such view:

$$F : D \times X \rightarrow Y, \quad (1)$$

that is strategies planning process means an application of decision set $D = \{D_0, D_1, \ldots, D_n\}$ to the set $X_0,\ldots,X_{n-1}$, that formally is considered as Cartesian product of sets $X \times D \rightarrow Y$, where Y – the set, which defines ACS at moment of goal achievement.

The transition of RTS from its initial state to goal is a sequence of state's transformations and has view:

$$\begin{bmatrix} x_0^0 \\ x_0^1 \\ \ldots \\ x_0^{n0} \end{bmatrix} \Rightarrow \begin{bmatrix} x_1^0 \\ x_1^1 \\ \ldots \\ x_1^{n1} \end{bmatrix} \Rightarrow \begin{bmatrix} x_2^0 \\ x_2^1 \\ \ldots \\ x_2^{n2} \end{bmatrix} \Rightarrow \cdots \Rightarrow \begin{bmatrix} x_n^0 \\ x_n^1 \\ \ldots \\ x_n^{nn} \end{bmatrix} \equiv \begin{bmatrix} y^0 \\ y^1 \\ \ldots \\ y^n \end{bmatrix}$$

It correspond to real situation, when in process of generation and execution of solution there is an evolution of RTS states.

However, the mentioned sequence of changes describes not only the problem-solving process, but also the dynamics of system's changes in time. On strategies planning the system's state changes in active mode, so at every step the strategies planning may change the characteristics of RTS. If to consider the sequence of actions on strategies planning there is need to define the function (vector) of problem-solving $\vec{D} = \{D_0, D_1,\ldots, D_{n-1}\}$.

Therefore, the application of decision D_i for every step of ACS functioning leads to transformation of matric column $X_i^j \rightarrow X_{i+1}^j$ of RTS states.

$$\begin{bmatrix} x_0^0 \\ x_0^1 \\ \ldots \\ x_0^{n0} \end{bmatrix} * D_0 \Rightarrow \begin{bmatrix} x_1^0 \\ x_1^1 \\ \ldots \\ x_1^{n1} \end{bmatrix} * D_1 \Rightarrow \begin{bmatrix} x_2^0 \\ x_2^1 \\ \ldots \\ x_2^{n2} \end{bmatrix} * D_2 \Rightarrow \cdots \begin{bmatrix} x_{n-1}^0 \\ x_{n-1}^1 \\ \ldots \\ x_{n-1}^{n-1} \end{bmatrix} * D_{n-1}$$

$$\Rightarrow \begin{bmatrix} x_n^0 \\ x_n^1 \\ \ldots \\ x_n^{nn} \end{bmatrix} \equiv \begin{bmatrix} y^0 \\ y^1 \\ \ldots \\ y^n \end{bmatrix}$$

Let's also notice, that the case of adaptive strategies planning needs to take in account the effect of "third side", for example of external world objects or rival, which affects (positively or negatively) to the strategies planning process. From one side, the effect of external workspace may be direct and when, to take in account it's existence and affection to strategies planning process we need to introduce the additional factor S of external WS states, containing the objects's set $S_i = \{s_i^0, s_i^1, \ldots, s_i^m\}$, with index i for discrete states of external WS:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2021 doi:10.20944/preprints202101.0028.v1

$$\begin{bmatrix} x_0^0 \\ x_0^1 \\ \dots \\ x_0^{n0} \end{bmatrix} * \begin{bmatrix} s_0^0 \\ s_0^1 \\ \dots \\ s_0^{m0} \end{bmatrix} * D_0 \Rightarrow \begin{bmatrix} x_1^0 \\ x_1^1 \\ \dots \\ x_1^{n1} \end{bmatrix} * \begin{bmatrix} s_1^0 \\ s_1^1 \\ \dots \\ s_1^{m1} \end{bmatrix} * D_1$$

$$\Rightarrow \dots \begin{bmatrix} x_{n-1}^0 \\ x_{n-1}^1 \\ \dots \\ x_{n-1}^{n-1} \end{bmatrix} * \begin{bmatrix} s_{n-1}^0 \\ s_{n-1}^1 \\ \dots \\ s_{n-1}^{m-1} \end{bmatrix} * D_{n-1} \Rightarrow \begin{bmatrix} x_n^0 \\ x_n^1 \\ \dots \\ x_n^{nn} \end{bmatrix} \equiv \begin{bmatrix} y^0 \\ y^1 \\ \dots \\ y^n \end{bmatrix}$$

The other way is in introduction of functional dependence for particular acts of strategies planning of workspace's states:

$$F : D(S) \times X \rightarrow Y,$$

and, correspondently:

$$\begin{bmatrix} x_0^0 \\ x_0^1 \\ \dots \\ x_0^{n0} \end{bmatrix} * D_0 \left(\begin{bmatrix} s_0^0 \\ s_0^1 \\ \dots \\ s_0^{m0} \end{bmatrix} \right) \Rightarrow \begin{bmatrix} x_1^0 \\ x_1^1 \\ \dots \\ x_1^{n1} \end{bmatrix} * D_1 \left(\begin{bmatrix} s_1^0 \\ s_1^1 \\ \dots \\ s_1^{m1} \end{bmatrix} \right)$$

$$\Rightarrow \dots \begin{bmatrix} x_{n-1}^0 \\ x_{n-1}^1 \\ \dots \\ x_{n-1}^{n-1} \end{bmatrix} * D_{n-1} \left(\begin{bmatrix} s_{n-1}^0 \\ s_{n-1}^1 \\ \dots \\ s_{n-1}^{m-1} \end{bmatrix} \right) \Rightarrow \begin{bmatrix} x_n^0 \\ x_n^1 \\ \dots \\ x_n^{nn} \end{bmatrix} \equiv \begin{bmatrix} y^0 \\ y^1 \\ \dots \\ y^n \end{bmatrix}$$

Therefore D_i as the strategies planning act depends of state of external workspace objects.

The difference of both the ways isn't very expressive, but the explanation can be inequal. In the first case the strategies planning system directly interacts to WS and such interaction leads to changes on RTS states, and therefore the act of strategies planning relates to system's state, affected by WS. For the second case, the planning act depends of WS state and must take in account its effect during the definition of decision's procedures (strategies), and decision executor transforms the state of RTS being the WS-dependent.

Therefore, the ACS goal at stage of strategies planning for the given task is in determination of ordered set (vector) $\vec{D} \subset D$, as a set of problem-solving acts, implementing the transition of robot's ACS from initial state X_0 to the goal Y as to expression $F : D \times X \rightarrow Y$.

The importance of adaptive problem-solving arises in case of essential changes on conditions of decision implementation. For case of robot's static workspace, the goal Y, as a state of RTS, is reached by application of possible action's set $\vec{D} = \{D_0, D_1, \dots, D_{n-1}\}$, which tranfers the systems form initial state X_0 to the goal $X_{n-1}=Y$. The set of selected actions $\vec{D}$ is considered as a decision plan.

For static WS there is developed the initial decision plan $\vec{D}_0 = \{D_0^0, D_1^0, D_2^0 \dots, D_{n-1}^0\}$, where the particular decision acts (strategies) are directly connected, and application of local problem-solving act D_i to the current state X_i will transfer the system to the state X_{i+1}, which is correspondently goal for decision act D_i. In it's turn the state X_{i+1} is initial for new state D_{i+1}, which will transfer system from state X_{i+1} to X_{i+2} etc.

For case of WS dynamic state the problem-solving system be application of decision acts D_i can transfer system to such state X_{i+1}, which can be insufficient to implement ac-

tion D_{i+1} and will acquire the additional decision acts D'_{i+1}, D'_{i+1}, etc. Therefore, the changes of WS will lead to indetermination of possible problem solving tools.

3. Conclusion

Action planning for robotic systems can be mostly divided into transportation and manipulation tasks. For transportation problem in static workspace, solution can be effectively reached by computation methods. If system has history (of knowledge) of problem solution, AI-methods become more actual, especially for application of neural networks and genetic algorithms. More complex transportation problem appears for 3D systems of warehouse type with vertical shelves or for container terminal for seaport. While with different scale the latter tasks look very similar to 3D problems for manipulators of robots, especially of manipulator makes actions in narrow space of obstacles. Here, knowledge of system becomes more critical and helps to solve the number of problems.
But for conditions of dynamic space, situation is more complicated. The motion of other objects (equipment, robots, vehicles, humans) can easily stop execution of best calculated plan (or strategy – in terms of proposed paper). In this case system must be adaptive to solve its problems, or coming back to previous steps of solution, or to previous key points, or even to starting points [9]. In this case, robots, like humans, must be more logically intelligent (with experience in greater number of operator's schemes), while combining computations and AI-methods. Possibly, in this way we can find the combined solution for robots, which try to be intelligent.

References

1. Tsymbal, O., Bronnikov, A., Mercorelli, P. Decision-making models for Robotic Warehouse. Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Virtual Meeting, 2020, June 24-26, P. 546-551.
2. Mikhailov, E., Remenyuk, B. "Optimize the placement warehouse transport system", Journal of Electrotechnic and Computer systems, 2015, No. 18 (94), pp. 60–64.
3. Kerak, P. Novel trends in the intelligent manufacturing systems. Proc. Of 8th International Baltic Conference "Industrial Engineering", 19-21 Apr., 2012, Tallinn.
4. Red'ko V. "Interaction between learning and evolution in population of autonomous agents", 2013, *Computing*, Vol. 12, Issue 1, pp. 42–47.
5. Eiter, T., Wolfgang, F., Leone, N., Pfeifer, G. A Logic Programming Approach to Knowledge-State Planning: Semantics and Complexity. ACM Transactions on Computational Logic, 2004, vol. 5, pp. 206–263.
6. Tsymbal, A., Bronnikov, A. Decision-making in Robotics and adaptive tasks. Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012), 2012, pp. 417-420.
7. Nevliudov, I., Tsymbal, O., Bronnikov, A. (2018) "Intelligent means in the system of managing a manufacturing agent", Innovative Technologies and Scientific Solutions for Industries, No. 1 (3), pp. 33–47.
8. Nevliudov, O. Tsymbal, A. Andrusevitch, V. Gopejenko. Intelligent Decision-Making Support for Flexible Integrated manufacturing – Riga: ISMA, 2020. – 390 p.
9. Nevliudov, I., Tsymbal, O., Bronnikov, A. "Intelligent means in the system of managing a manufacturing agent", Innovative Technologies and Scientific Solutions for Industries, 2018, No. 1 (3), pp. 33–47.
10. Vacic ,V., Sobh, T. Vehicle routing problem with time windows, Computing, 2004, Vol. 3, Issue 2, pp. 72–80.
11 P. Mercorelli and et al., "A model predictive control in robotino and ist implementation using ros system," in 2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARSITEC), 2016, pp. 1–6.
12 P. Mercorelli, T. Voss, D. Strassberger, O. Sergiyenko, and L. Lindner, "Optimal trajectory generation using MPC in robotino and its implementation with ros system," in 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017, pp. 1642–1647.