Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Complex automation of technological processes with involving event model in feedback control scheme

Alexander A. Ambartsumian*, Dmitry L. Kazansky**

* *Institute of Control Sciences, Russian Academy of Science(s) Moscow, Russia; (tel.: +7-495- 334- 87-89, e-mail: ambar@ipu.ru)*

** *PLCSystems Company, Moscow, Russia (e-mail: dkaz@plcsystems.ru)*

Abstract: This paper introduces a method of technological process control for manufactures with continuous structure. This method assumes using of three-leveled event logical manufacture model. This model allows generating the technological process structure on the basis of the current and targeted structure formal description analysis. It provides the subsequent extraction of required structure from knowledge base or interactive construction by means of involving the operator into the process of creation the required structure. The next control step is the realization of the chosen structure and automatic adjusting the necessary settings for involved technological aggregates. Such an approach provides a possibility of the system protection from the personnel's errors by the means of restricts insertion on his actions in the form of a dialogue scheme. The work is supplied by the RFFR RAS fond. The grant №06-08-01619-a.

1. INTRODUCTION

Technological process execution in some kinds of discrete continuous manufactures (like oil extraction, initial oil treating, oil and gas storage, oil or gas transport) is supplied by the treating of the material parameter maintenance and from the other hand by creation of flows of such materials, following through corresponding equipment. Usually the functions of regulation, security alarms, signalization and blocking are realized in the automation systems of these ('flow-type') manufactures, while the forming function of the stream structures is the task of the human-operator. Because of this a very low automation level is supported and in the other hand operator is still remains over all tasks and continue to be the initial and controlling unit of the control chain. At the same time the basic restrictions for the operator's actions still are the job instructions and regulation prescriptions. As a result it cause increasing of human factor role in control process and decreasing of effectiveness and security of these manufactures.

Modern complicated 'flow-type' manufacture processes are characterized by different process flows, which are combined into one technological process, which is heterogeneous by its function and structure. Developing of the 'factory floor'automation systems with complex control for these manufactures, where both material stream parameters control functions and control function of the stream structure are included, is the actual problem now.

There are some publications by [H. van Brusel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters], [Edgar Chacon, Isabel Besembel, Jean Claude Hennet], which are present multi level models based on holons and agents for the control in distributed flow-type manufacture systems. But in these publications not takes into account process control scheme and personnel's role.

In this paper authors suggest methodology of the complex management of technological process and equipment in control system based on imitation of the human-operator actions, which manually controls technology. Author uses term "complex management" in the sense of coordinated with all aspects of technological process management.

Methodology is considered onto logical framework level: models, control scheme, and basic procedures.

It's is based on following hypothesis about human-operator actions at manual management of flow type technology:

- Based on the real situation and technological aim, human-operator is choosing one from the list of the potential scenarios (scenario is the set of rules for running technological processes determined by technologist before);
- By using this scenario, human operator pass through technological structure and tune it up for current requirements (preliminarily find out concrete state possibility);
- Streams starting;
- Tune stream parameters for technology requirements and then supervise both structure and parameters of the processes and equipment for the given value correspondence.
- In the case of the achievement the goal or anomaly, correction or choice and next scenario applying is occurred.

It is necessary to obtain all components mentioned in the above hypothesis for human operator actions algorithmization. The three level event model of automation manufacture is offered to use as this model (EM – events model). This model has been described by papers

[Ambartsumian A,A., Kazansky D.L. 2001, Ambartsumian A.A., Potehin A.I. 2004] and extended in this paper by dialog schemes and process monitoring tools. These extensions required control scheme modernization.

2. COMPLEX EVENT MODEL OF TECHNOLOGICALLY CONTROLLED OBJECT AND PROCESSES

Complex event model of technologically controlled object and processes consists of the following:

A model of *TN* "Technological Network" that contains an executive mechanisms models (*A*) and models of material links (*R*).
Technological process model (*TP*) – technologically required configurations of machinery (*TN* fragments, blocks, refills, etc. that provide execution of certain technological tasks).

Job schedule (*TS)*-active technological scenario scripts.

Model of the Technological Network defines the production structure and consists of a variety of machinery and links. It shows how aggregates (valves, pumps and tanks) are connected to each other through material flows and provides tracking of the material movement and its properties' change while processing *TP*. Technological network represented as an oriented graph $TN= <A, R>$ which nodes $A=\{a_i|i\in I_A\}$ defines different producing machinery and a set of edges $R=\{r_{ij}|i,j\in I_A\}$ defines material links which connects product inputs and outputs of different machinery.

Aggregate model (components of *A* set) defines the behavior and properties of minimal atomic parts of technological network that can change parameters of material or properties of a flow. Aggregate model is a set of $a_i =< H_i,Z_i, U_i, X_i, LCA_i>$ where: H_i,Z_i is a set of material inputs and outputs for aggregate, U_i,X_i- aggregate commands and its state indication Y_i (connection between X_i и Y_i is set on defining model of a certain aggregate). $LCA_i = <U_i, X_i, Y_i, \delta_i, \lambda_i, D_i>$,is a life cycle that is described with a finite state machine which defines connection between states of an aggregate and their change conditions (function of the X_i). Transition function δ_i, defines rules of aggregate's state change form current y_t to a new state y_{t+1} it depends on the command $y_{t+1} =\delta\ (y_t, u_t)$.Outputs function λ_i returns values of info outputs from current aggregate state and a command sent to aggregate : $x_t=\lambda(y_t, u_t)$. New to aggregate model is a set of dialogs D_i which defines the options of human and aggregate collaboration. Each dialog sets personnel messages and available reaction options (the model of the dialog will be represented further).*TN* model has the following properties: on the one hand, it reacts to command-events with state change and confirmation events generation and, on the other hand, it provides the special procedures to define implementation of a certain process on current *TN* state; calculate desired control action on certain aggregate on its set up; forms required condition of process integrity.

Technological Process model *TP* in scheme of control represents state of real processes and emulates their execution by changing the state of model's lifecycle; defining local technological aim; defining indicators of life cycle's steps completion and cases of process integrity which are reliable for process monitoring. Formally, the model is represented by a set $TP_j = <A_j, R_j, CS_j, LCP_j>$, where A_j and R_j are subsets of aggregates and material links that are used in process TP_j. A quantity of *TP* configurations $CS_j = \{cs_s|s\in I_{cs}\}$, where each configuration is defined by the following set $cs_S = < ms_S, m\varphi_S, m\psi_S, lm_S>$. Here $ms_S, m\varphi_S, m\psi_S$ as well as in [Ambartsumian A,A., Kazanskiy D.L. 2001] are cortege of aggregates commands, implementation cases and readiness functions. They defines a set of aggregates for TP_j that are used in configuration cs_S, sequence of their state change into the desired state, conditions (on each aggregate) of its ability to participate, and readiness for work in process TP_j. Cortege of customization is formed on process's model definition stage, when it's clear when and how certain configurations of process are technologically available and required. Corteges are convenient to represent with a table with column headers containing aggregates unique names and rows with their states and sequence for each process configuration, lm_s is a set of control case sequences for process configuration monitoring.

For each process, configuration rules of technology defines critical events in which occurrence is identified by certain cases. Those cases are the signal for executing certain control actions on TP. Event models contain the sets of those cases and bound actions that are defined in sequential case form: *case* $\rightarrow$ *action*. Case here is a Boolean expression which arguments hold parameters and variables of TP machinery states, phases and states of TP models, and historical knowledge on process execution. And *actions* are control commands for aggregates, process models or personnel.

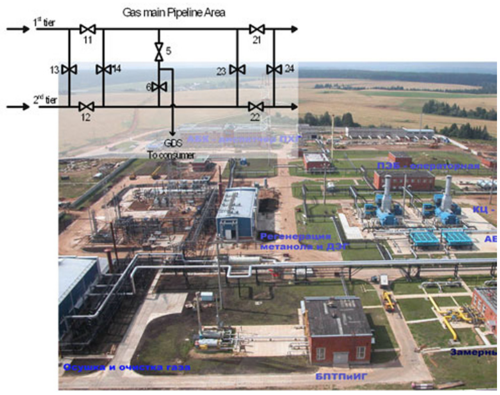

Figure 1 Line area of gas main pipeline

A set of sequence records that describes parameter control for certain TP in certain configuration is called private monitoring list for TP – ml_s Figure 1 is an example of fragment of gas transporting network on line area of gas-main pipeline which holds the following material flows:

a_{11}, a_{21} a gas flow on the first stream of main pipeline it provides technological process of the main gas transport; a_{12}, a_{22} a gas flow on the second stream of main pipeline provides the same task; a_5, *GDS* –a flow to gas distribution station –

provides a process of gas distribution to local consumer from the main pipeline

Table 1 is a sample of customization corteges for TP structure: "Line 1 – gas transport over stream 1" for two configurations: cs_1 – exclusion of stream 1 from TN and cs_2 – inclusion of stream 1 to TN; the analogy is for "Line 2" and "Gas transport to GDS". There we connect process matrixes ms_1/ $m\varphi_1$/$m\psi_1$, ms_2/ $m\varphi_2$/ $m\psi_2$ and ms_3, $m\varphi_3$, $m\psi_3$

Aggregates positions required in each configuration are set by names of aggregates' states that were defined in lifecycle of their models. Here, for example, y_1 identifies closed position of a valve and y_3 the opened one. For implementation of the selected configurations it is required to change state of all aggregates of process to the desired state. Aggregates marked with ~ are not needed, they can remain in any position.

Table 1 Line 1 – gas transport over stream 1 (*TP₁*)

a	11	12	13	14	21	22	23	24	
MS	y_3	y_1	y_3	y_1	y_3	y_1	y_1	y_3	Cl.
	~	y_3	~	~	~	y_3	~	~	Op.
MΦ		P_{LC2}< P_{nor}				P_{LC2}< P_{nor}			
MΨ	x_3	x_1	x_3	x_1	x_3	x_1	x_1	x_3	
		x_3				x_3			

Table 2 Line 2 – gas transport over stream 2 (*TP₂*)

a	11	12	5	6	
MS	~	~	y_3	y_1	From 1st str.
	~	~	y_1	y_3	From 2nd str.
MΦ	P_{LC1}>P_{nor}				
		P_{LC2}>P_{nor}			
MΨ			x_3	x_1	
			x_1	x_3	

Table 3 Gas transport to GDS

a	11	12	5	6	
MS	~	~	y_3	y_1	From 1st str.
	~	~	y_1	y_3	From 2nd str.
MΦ	P_{LC1}>P_{nor}				
		P_{LC2}>P_{nor}			
MΨ			x_3	x_1	
			x_1	x_3	

Table 4 contains private monitoring list for all configurations and is intended to be added to main monitoring list. Reactions here are represented by identifiers of active scenario scripts which are called on condition occurrence. For example, action «*TS*14» is for calling active script that makes redistribution of gas flow between streams by preset pressure on input and output of an area; and action d_{25}(*LC-1-1*) executes dialog d_{25} with parameters of a process.

Table 4. Monitoring list for stream 1 processes

S	Cases	Actions
1	P_{min}>P_{LC1-11}>P_{max};	*TS* 14(*LC*-1-1)
	P_{min}>P_{LC1-21}>P_{max};	*TS* 14(*LC* -1-1)
1,2	Q_{GDS}>Q_{plan1};	d_{16}(*LC* -1-1)
	Fire - GDS;	*TS* 26(*LC* -1-1)
	t>t_{normal};	d_{25}(*LC* -1-1)
2	P_{min}>P_{LC1-12}>P_{max};	*TS* 14(*LC* -1-1)
	P_{min}>P_{LC1-22}>P_{max};	*TS* 14(*LC* -1-2)

Life cycle LCP_j of process TP_j (see fig. 2) is a finite state machine $LCP_j =< E_j, S_j, W_j, \delta_j, \lambda_j, s_0 >$, which states $S_j = \{s_l | l \in I_{Sj}\}$ are steps (phases) of TP execution: "unclaimed" state (s_0) is a state of checking of process implementation on current state of TN;(s_1) is a "set up" phase for changing aggregates state to desired to run TP_j; (s_2) "execution" of TP_j with preset structure and parameters; (s_3) "execution with deviations" (reconfigure); (s_5) crash finish or normal finish; (s_4) typical configuration of TP. $E_j = \{e_k | k \in I_{Ej}\}$ is a set of input symbols – input events that controls lifecycle LCP_j state change. The main events here are start event (automatic or manual); check of potential readiness of TN to execute TP_j - $e_i ::= (\Phi_j = 1)$; active state change event TP_j - $e_q ::= (\Psi_j = 1)$ that confirms the fact of configuring *TN* to execute TP_j; event of process parameters deviation from normal $e_r ::= (\Psi_j = 0)$ - execution of "flow" alarms and blocks; event of "unmounting" (killing) process. $W_j = \{w_i \mid i \in I_{Wj}\}$ is a set of output symbols (events) that are spawned by process as structural part that characterize TN state: command of execution sub-processes; message about process's reconfiguration; queries to personnel; archive messages; execution of flow blocks and alarms.

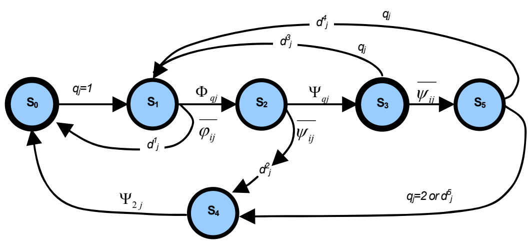

Fig. 2 Lifecycle of a process

Functions δ_j and λ_j bounds input events, states to outputs (for example with transition graph fig. 2).

3. ACTIVE TECHNOLOGICAL SCRIPTS

Aimed production functioning is defined by current plans and is provided by *technological orders* (reglaments). Those orders define a sequence of execution and finishing process, monitoring lists, process state change in the view of current situation or task change. Sequenced operations are naturally modeled by weighted oriented graphs – *technological scripts* (*TS*) and their interpretation can replace personnel actions if they are strictly defined. *ATS* are diagrams which represent plans of achieving certain technological aims as a unity of steps, their sequence and execution cases. Formally, *ATS* is defined as a set *TS*=<*STR, INST, LCS, ml*>, where *STR* – script structure that is set up by transition graph (nodes are equal to instructions and edges are sequences of execution).

There are some types of graph nodes (actions, alternative branching, beginning and end, parallelism and cycle). Nodes (type action) represent next step (position) of script. The Step can have pre-case and post-case. Pre-case means that actions in this step can be executed only after that case is true. Here a sequential form of case definition is used "**IF** <case> **THEN** <command>"; <case> is a logic expression which arguments hold TP parameters, machinery states, phases an states of TP models, and stored knowledge about process execution. In this case the way step is a sequence of TP control commands. Command list can contain control commands for state of executive mechanisms, commands for reconfigure of TP model, command of calling sub process for execution, commands for personnel manual actions. If the step has a single instruction for personnel it's marked as a dialog. And if the instruction requires execution confirmation then case has a call to dialog window with a user's action confirmation in post-case. *LCS* – (Life Cycle of a Script) is defined by a set of states and rules of their change. *ml* –monitoring list is a set of case sequences that controls script execution.

TS is used for execution of a certain control goal. For each task a new script is implemented which step by step builds up the necessary structure and working mode for production object on the basis of previously defined TP models control. Script makes a set up of necessary working modes; it controls personnel for execution of non-automated (manual) actions, and gives a hint to an operator to choose an alternative.

4. MODEL OF DIALOG BETWEEN HUMAN AND AUTOMATED SYSTEM

In the models of all 3 levels the special construction is introduced – the scheme of dialog between automation system and a human-operator, which becomes more intense in accordance with logic of *LCi* or/and *LMi*. The dialog scheme $Sh = < F, Q, Ans, ans0>$ defines an order of interactions between user and system, describing dialog shape F, queries and message to user Q and probable variants of user's reaction of scheme *Ans, ans0*. In the analysis stage of a project a dialog of certain kind and shape shall be defined for each operator's decision case. A dialog describes the current situation, defined current aim and queries reaction of a human operator. Dialog format is set for each certain case by a dialog model that defines info messages for an operator, command instructions, answer variants, an available reaction time and default answer.

The following types of dialogs are offered: 1- queries for the fact of event; 2 – enter a scale and value of parameter; 3 - selection of devices and variant of execution process; 4 – selection of variant of script; 5 - coordination of process, actions of an operator and technical personnel during the control of an object as a whole. The dialog of the 5-th type is possible as a dialog with agent, which solver is based on ontology of object domain; it allows executing coordination of the processes using logical analysis of experience accumulated in ontology.

There are dialogs, used in model of aggregates: query for aggregate position, query for visual aggregate parameters; description of aggregate installation in required position, command for manual control and confirmation of command execution. There are dialogs, used in model of process: in monitoring list, for information about changes in process configuration , labor–rent of blocs and alarms: in table MΦ for query of state of non-automated aggregates, puts, parameters of process, and for checking of conditions of start of process configuration. In model of active scripts dialogs can be implemented in all nods (action or bifurcation).

5. CONTROL SCHEME ON THE BASIS OF EXTENDED EVENT MODELS

In the suggested models, their functioning and control is defined in structure of TOC.

The control of production in a complex event model is determined in scheme (see fig. 3), based on stage interpretation of technological script and on commands executions (stipulated in stage) of starting and stopping (cancel). The EM-graphs view doesn't depend on specific TOC, it changes the dimension and edges weight only. This allow to build the control scheme as package of procedures to process graphs (life-cycles and scenario graphs) and to forming by the results of their execution the control commands or dialog with personal requests. These procedures are included in Autooperator (AO) module, on fig. 3 - this is Autodispatcher.

At every moment of time *t* the *TP* is separated into active SAP_t and passive SPP_t. subsets, as much as multitude of all scripts of *TS* is separated into active ATS_t and passive PTS_t. Functioning of *EM* is defined in discrete moments of time t = 1,2,3,… Every moment of time *t* model receives U_t, X_t and P_t –state (position) of aggregates and flow parameters; and a set of events $E_t = \{e_k | \ k = 1,...n\}$ take(s) place that is provided by real aggregates, automaton system components and personnel (here we map all system events in E_t for short). State of automated production at a moment *t* is a unity of *TN* state and state of active scripts subset $SAM_t=<STN_t, ATS_t>$.

State of *TN* in a moment *t* is a cortege $STN_t = < y_i | \ j=,...m>$ of the state of all aggregates, where y_i- state of aggregate a_j at moment *t.*

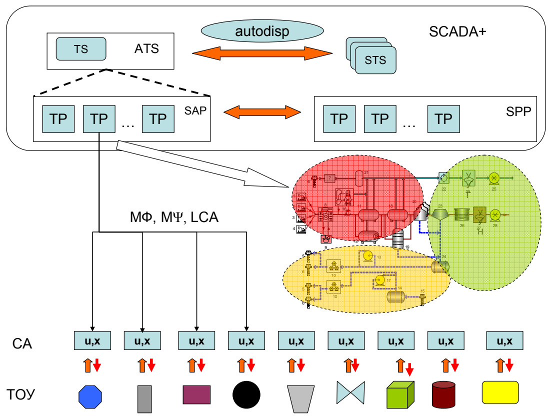

Fig. 3 Control scheme

Functioning of *EM* production model is based on transforming the flow of input events E_t (active states of appropriate models lifecycle) $STN_t \times E_t \rightarrow STN_{t+1}$; $SAP_t \times E_t \rightarrow SAP_{t+1}$; $SPP_t \times E_t \rightarrow SPP_{t+1}$; $ATS_t \times Et \rightarrow ATS_{t+1}$; $PTS_t \times E_t \rightarrow PTS_{t+1}$ into new states and events and creating by result $E_t \times STN_t \times SAP_t \times SPP_t \times AST_t \times PTS_t \rightarrow E_{t+1}$ a new event cortege E_{t+1}. Those transforms are made in a cycle based on the event flow $E_0, E_1, E_2, \ldots$, by rules of transformation of transition functions for finite state machines. Therefore, the functioning of event structure model and its processes is as follows: event flow $E0, E1, E2, \ldots Et \rightarrow \propto$ sequence SAT_0, SAT_1 ,...., SAT_t of active processes, sets of STN_0, STN_1 ,...., STN_t aggregates state, and active script set ATS_0, ATS_1,…, ATS_t that provides the technological goals' achievement.

Depending on the real situation in TOC, autooperator selects a necessary strategy of TP guiding, that corresponds with stirring up the active script.(fig. 3 it is shown as transfer of TS from subset of passive PTS_t to subset of active ATS_t) and AOs interpretation of its steps. On each step the TP is initiated or stopped (canceled). During the start of *TP*, the AO stirs up instance of object of process model in TOC event model, and this instance is the real model of *TP* (fig.3 it is shown as transfer of TS from subset of passive SPP_t to subset of active SAP_t). Attributes of *TP* model: structure, state of life cycle, parameters of flow, state of block and alarms' function, and automatic regulation; reflect all types of TP in all stages of life cycles (checking of implementation, start, work in current process configuration, dismantling).

The fragment of TN fig.1 shows an example of EM functioning. All aggregates function normally; flows of gas transportation are formed on both lines SAP_t= {TP_1, TP_2, TP_3}; SPP_t=∅. An operator controls it with HMI. In case of an alarm situation when the 1st line should be closed, the operator decides to activate an active script for this goal, and fires the command; TS_1.Script automatically checks the ability of achieving the goal and instructs the operator on sequence of organizational works. In the position of automated process control the command is executed "build TP_1 - flow r_1: a_{13}, a_{12}, a_{22}, a_{24}". An autooperator calculates implementation function $\Phi_1=(P_{11}<P_{const1})(P_{21}<P_{const2})$ with $M\Phi_1$. As all aggregates and other processes allow the flow r_1, so Φ_1=1 and *EM* gives the operator KA_1 (if Φ_1=0 then *TP* execution is not allowed and the operator is warned about it); LCP_1 is in s_2 state; start process continues (automatically or manually if executive mechanism is not automated). Now the aggregates are set, their models will change to state defined in MS_1,events; in this case goes from layer *TN* into $SAP_t \Leftrightarrow SPP_t$, that is the reason for $\psi_i = x^1_{11}\cdot x^3_{12}\cdot x^3_{13}\cdot x^1_{14}\cdot x^1_{21}\cdot x^3_{21}\cdot x^1_{23}\cdot x^3_{24}$ is equal to 1 and LCP_1 changes to s_3 state. After that, an autoperator changes TP_1 from *SAP* to *SPP* and MMI, operator will receive the appropriate information about macrostate [SAP_{t+1}= {TP_2, TP_3}; SPP_{t+1}= {TP_1}.] that is equal to set up TOC to achieve the goal. State of $s(TP_1)$=1 is passive and $s(TP_1)$=2 is active. The sequence of state change in *EM* is as follows:

SAP_t= {TP_1, TP_2, TP_3}; SPP_t=∅ – *initial network state;*

p_k: $U(TP_1)=1$; – *command of configuration change for* TP_1 *in k scenario position;*

p_{k+1}: $U(TP_3)=2$; – *command of configuration change for* TP_2 – *change to 2nd line;*

SAP_{t+1}= {TP_2, TP_3}; SPP_{t+1}= {TP_1} – *the result state of the net.*

Activation of a real process is made with step by step set up of all of its aggregates for states defined in TP model. *Steps are performed according to current TN deviation from the goal of TP.* Set up procedure based on TN analysis data is run by AO.

The AO behavior is defined so that all functionality of control scheme is directed to serve queries of TP, which means to provide dynamics of TP lifecycle. TP is built around the material flows and shall provide their functioning according to tactical production goals. So control scheme is executed as cycle procedure.

1. Low level automated control systems works, structure of flows is defined by an active process, flows function under control of regulators, blocks and alarms, and a set of actual events is formed.
2. The state of event model of *TN* is calculated, The state of active and passive process and scripts is analyzed and if correction of active and passive *TP* sets is not needed then go to 1 else go to 3.
3. The type of correction is defined, the deviation of current TN structure from desired is calculated .If a situation contains deviation then an operator selects new TS or correction option of TN state. The options are: start/stop of active process, start/stop of attached process, reconfigure of active process; a set of active and passive TP is refreshed; cycle repeats from 1.

Notice that the aim of configuration control is a built of a certain *TN* (or its fragment) structure, so the description of the desired structure is the aim definition. Content of the aim has definition of: TN structure, the desired components state, flow parameters and automatic control procedures variables (blocks, alarms, PIDs, etc., if they are defined for the *TN* fragment). That is why this type of control is presented as a complex.

6. CONCLUSIONS

The article offers a three-level production event model and scheme of control based on this model as a flow of events and commands in the following set: <technologically controlled object, automation system, SCADA, event model, human-operator>. Mainly, scheme of control is the basis in the concept of control system and it defines its functions and view.

Let us formulate view of automation system with complex management

1. Layer of the procedures realizing constantly operating restrictions and procedures of regulation of parameters (automatic regulators).
2. Event model layer as data structure storing technological network and other models.
3. Layer of the graph processing procedures (technological network, aggregates and processes lifecycles and scenarios graphs) forming control commands or personnel dialog requests.

The model provides protection of technological processes and production from errors of personnel due to introducing restriction for personnel's actions as dialog schemes.

This approach allows transition of a number of functions from a human-operator to an automated system. It's shown at fig.3. Functions 1-4: process state control, events reaction, TP control and technology organizing are down leveled to automated control system level different from traditional systems where these functions are above the line and at upper levels of control.

Control and action restriction mechanisms provide some types of dialog with personnel, which define its role in control at a certain moment. Dialogs are equal to level of decision making and they allow addressing queries according to personal reliabilities and initiate human participation in control only when it's required by technology and personnel papers. Thus, dialog schemes built –in active scripts, process and aggregates models define the role of a human in a chain of technological process control at a certain moment of the call.

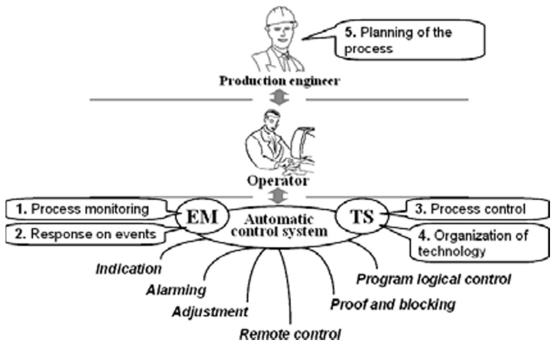

Figure 3 Functions layering.

Control functions distribution between personnel under control of active scripts allow to raise the efficiency of control and technological processes' safety due to restriction of human actions in control, framework to actions that could be applied in a certain situation at a certain moment of time.

REFERENCES

H. van Brusel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters, *Reference architecture for holonic manufacturing systems:* PROSA, Computers in Industry 37 (1998) p.255–274.

Edgar Chacon, Isabel Besembel, Jean Claude Hennet. *Coordination and optimization in oil and gas production complexes.* // Computers in Industry №53; 2004. -p.. 17–37.

Ambartsumian A.A., Kazansky D.L. *Control of technological processes on base of event model. Parts I and II.* //Automation and Remote Control № 10,11; 2001.

Ambartsumian A.A., Potehin A.I. Development of control mechanisms of objects with continuous technology on base of channel event models. // Automation and Remote Control № 4; 2003.