

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



An Approach to Technological Processes Automation using Technological Coalitions Based on Discrete Event Models

Alexander Ambartsumian and Dmitry Kazansky
*Institute of Control Sciences, Russian Academy of Science
Russia*

1. Introduction

There is a common feature in all control systems in big plants that use complicated technologies. It is that their control systems usually consist of two parts. There is the automatic control part for simple control tasks and the supervisory control part for other tasks. Functional division between these parts is not stable and depends on how mature is the automatic control part. Of course the share of automatic control generally tends to increase, even though in the more complex cases this growth is checked by the lack of suitable algorithms. Thus supervisory (i. e. human) control will always remain necessary until algorithmization of all aspects of technological processes is completed. Today there are several methodological problems which remain outside the scope of existing approaches to control system design, which causes inherent flaws in the resulting algorithms. We propose to introduce a new approach that will address these issues, and we will explain our approach using a common type of industrial technologies called flow technologies. A few words about these.

A lot of different tanks, valves, pumps, separators, desalters (demineralizers), heaters, fractionators, coolers, petroleum gas flare system, boosters and some other devices (aggregates) are connected usually by pipes or conveyor belts and some flows of different substances pass through them while the properties of these substances undergo numerous changes. This is what in the present paper we are going to call “flow structure of technology” or just “flow technology”. We must stress that we will consider here only big plants which have a lot of types of technological equipment. We should notice that flow technologies are used extensively in oil and gas industry, chemistry (cryochemistry), power engineering, hydrometallurgy and in other types of heavy industries, manufactures and processing plants. We have here roughly outlined the areas where our research is likely to prove useful; those interested in the technological aspects of particular industries should refer to the literature on the subject. Flow technology is quite a common kind of technology – for example, we really can refer to processes of production and preparation in oil fields all over the world. Multi-field plants show appropriate example of this technology. It is a big illusion to think that the black liquid (the pure oil) is produced direct from wells. Usually only an oil-gas-water-sand mixture is really produced from the wells. Figure 1 below shows the structure of a typical oil-plant. There are several input flows from different oil-fields and

Source: Process Management, Book edited by: Mária Pomfiová,
ISBN 978-953-307-085-8, pp. 338, April 2010, INTECH, Croatia, downloaded from SCIYO.COM

there is one output for oil, one for gas, one for water (in most cases). We can refer an example of control concept (Chacon et al., 2004) but we developed another view and approach for control. Different combinations of equipment can be used for oil processing depending on raw oil characteristics (sulfur, water, dissolved gas, sand etc). There are a lot of combinations and, it is important, none of these combinations is stable for a long time. The period of stability is between 2 and 8 hours usually.

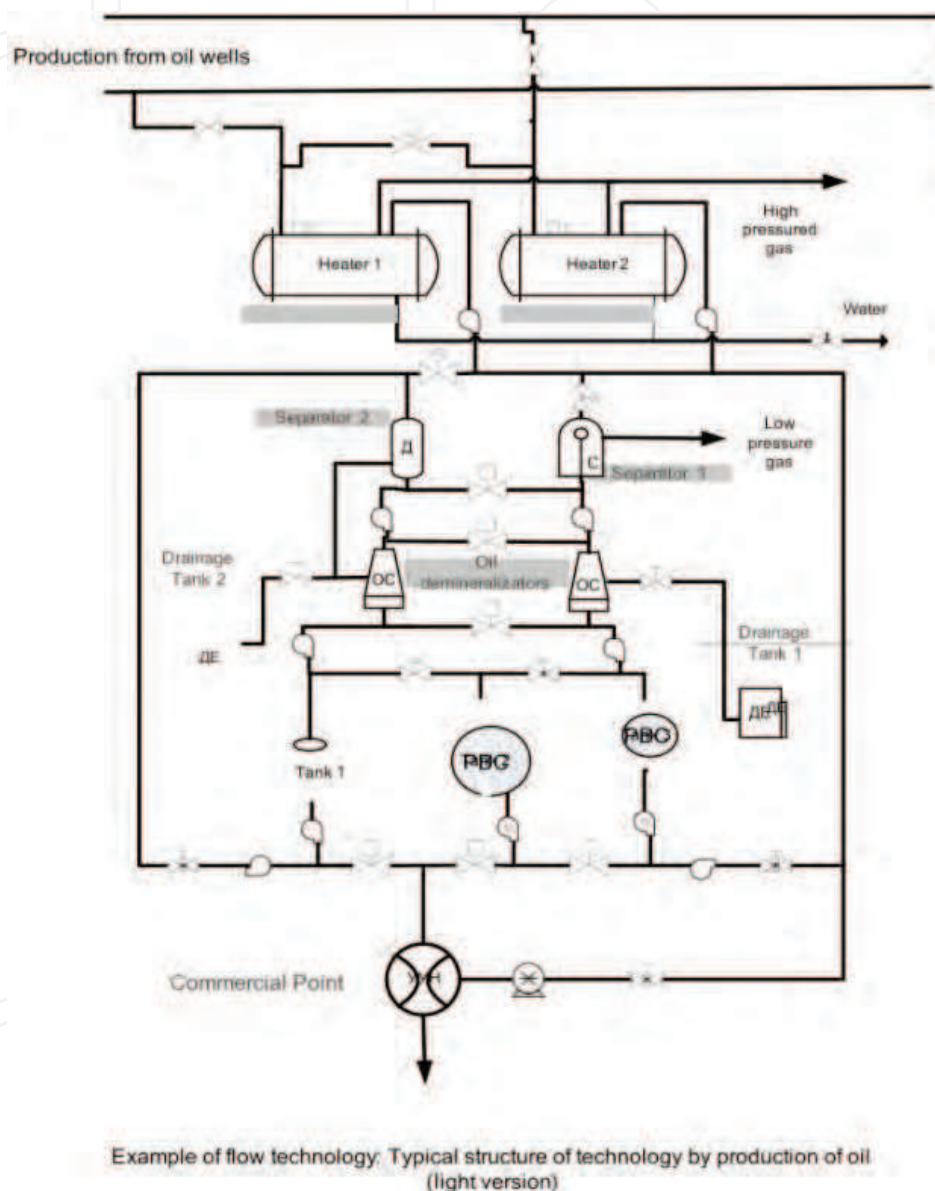


Fig. 1.

We'll focus on general flow technology control algorithms applicable to any of the above-mentioned industries. As we know there are three fundamentally different types of control functions in industrial applications:

-1st type - the local protections and alarms (simple one-step action, used for accident prevention. They are really very simple: "IF condition THEN action". The action itself is usually a one-step instruction to a single aggregate like open/close/switch_on/switch_off etc),

-2nd type - the local regulators (used to prevent certain parameters going above or below the required level. The regulator continually controls the position of a pump or valve at any given moment based on the values of technological parameters using P, PI or PID rules),

-3rd type - the multistep logical algorithms for group control (or MSLAs). MSLAs are used to determine how different hardware components in a flow technology context interact with each other.

All these types of control functions were invented and implemented in 20th century. The first two types are already completely formalized and automated. They are mature enough and this is confirmed by their worldwide use in real manufacturing. There are a lot of implementations and modifications of them and we won't discuss them any more in the present paper.

Our target is the third type of control functions, MSLAs. They are not popular in real manufacturing to the present day and we have difficulties with them. We'll try to answer why. MSLAs, although repeatedly tried, have not been found to perform satisfactorily in real life conditions because they can only work for a very short time before they have to be updated or modified. Hence there is an urgent need for change in design methods used for this type of algorithms. We should explain in somewhat greater detail what is in fact wrong with these algorithms and their design method.

In real life there are a lot of interfering factors which can disturb or upset the normal functioning of algorithms. We must consider different external and technological factors and parameters which can appear. All the three above-mentioned types of algorithms should ideally have the ability to register and process external changes, i. e. be adaptive. This is not an issue for type 1 and 2, there are easy and well-known ways to customize them. But there are no such easy ways for MSLAs. As a result it is typical for them to lose control ability.

Some time ago we commissioned a study the object of which was to find out how and to what extent MSLAs were actually being used in various industries. The results were not unexpected, though far from optimistic. We found that 45% to 60% of users stopped using MSLAs within the first 2 months, while after 3 months this figure rose to 75%. In all cases where MSLAs were no longer used an operator had to take over. When we asked what caused this change, the answer was quite simple. There are two reasons. The first reason has to do with the constantly changing properties of the substances being processed (raw oil, etc.). A single rigid algorithm simply ignores these changes, many of which are critically important. The second reason is changes in hardware introduced in the course of normal upgrades and maintenance of equipment (monthly or/and weekly) and the consequent small (but accumulating) changes in operating requirements. For example, when any part is replaced with a technologically compatible but slightly different part, the new part will interact with other devices in a more or less different way than the old part did. This fact of parts being constantly replaced with other non-identical parts (e.g., from another manufacturer or with slightly different specifications) is the basis for all (or nearly all) interfering factors for MSLA. Software developers didn't anticipate this and therefore made no provision for this in their algorithm. As a result the small changes in the characteristics and/or technological requirements of several hardware components can cause very deep changes and often require a full rebuilding of the MSLA. This problem doesn't destroy type 1 and 2 algorithms, but it is a serious problem for type 3 algorithms. We call this problem of increasing inability of the system to adapt to changing conditions "aging of MSLA" or "becoming out of date".

A few comments about “aging of MSLA”. Is the situation of “MSLAs aging” capable for improvement in principle? That is the question. MSLAs are usually designed before real launching of flow-technology. After a very brief period of real use MSLAs won't be able to provide adequate control any more because they will not be able to absorb the latest changes. We can of course completely rewrite the algorithm each time, but this is hardly an efficient approach. The problem is there isn't any regular (scientific) method and suitable tools for tracking changes and assimilating them in the body of MSLA. It is absolutely clear that the larger a MSLA is (i.e. the larger the number of steps it contains) the more vulnerable it is to external changes. These algorithms are destroyed by their big size.

The classical (and usual) way to describe the functioning of the algorithm today is to build an appropriate Finite State Machine (FSM) (e.g. a Moore machine or a Mealy machine) or a Petri Net (or to use another method based on these). After that it is necessary to supplement the control algorithms with some existing SCADA. It works, but for simple cases only. It appears that there is a fatal flaw in this method of design for MSLAs which results in algorithms that are unusable in the real world. Is it possible to separate the solvable and the non-solvable part of the problem for MSLAs? It seems at first sight that it is not a scientific problem. We did not expect to find many papers about MSLAs and about this problem. There is some truth in that, as we understood later, after looking in different sources (Wonham & Ramadge, 1988, Jennings et al., 2001, Yoo & Lafortune, 2002, Cassandras & Lafortune, 2008). Some other aspects were discussed during congresses and conferences (Golaszewski & Ramadge, 1987, Zambonelli et al., 1994, De Queiroz & Cury, 2000, Akesson et al., 2002, Gaudin & Marchand, 2003). There is not wide bibliography but where there is a will there is a way – and we started looking for some scientific solution to this problem.

This concludes the semantic introduction and the general description of the problem. We have determined the type of technology for investigation and sketched the main problems. Some results are below.

2. Different ways to use the Finite-State Machines in case of real control and in case of classical transformations of strings.

We have to spend some time and to draw the reader's attention to basic things. The ideal control situation as a general concept is the situation of informational interaction between two components. In this situation there is always a controlled component and a controlling component. The controlled component informs the controlling component about its events using a special pre-arranged alphabet. The controlling component receives information from the controlled component and sends the functionally defined, appropriate command; also using a special alphabet. The ideal control situation for MSLA is often described by means of FSM (e.g. Moore machine or Mealy machine). Let us look at a classical definition of a Finite-State Machine (FSM). As we know there is its classical definition, suitable for most applications:

$$A = \langle S, X, Y, \delta, \lambda \rangle, \text{ where}$$

S is a set of states.

X is an input alphabet (a finite, non-empty set of symbols).

Y is an output alphabet (a finite, non-empty set of symbols).

δ is function for states $(S, X) \rightarrow S$ (the state-transition function).

λ is function for outputs $(S, X) \rightarrow Y$.

We have to admit that the FSM is appropriate for cases involving string transformation. But real-life control situations are far more complex than that, and cannot be reduced to string transformation alone. There is difference between string transformation and real control situations. The nature of real control must allow the existence of additional external information of different types which can possibly affect the outcome - but hard-coded sets of transformation instructions ($X \rightarrow Y$) do not allow for any adaptive behaviour. As we see the classical definition of FSM serves only situation of transformation of strings. The classical definition of FSM allows to have only functionally defined commands. Availability of additional data and any special handling of them as basis for case of control didn't foreseen in the classical definition (Fig. 2).

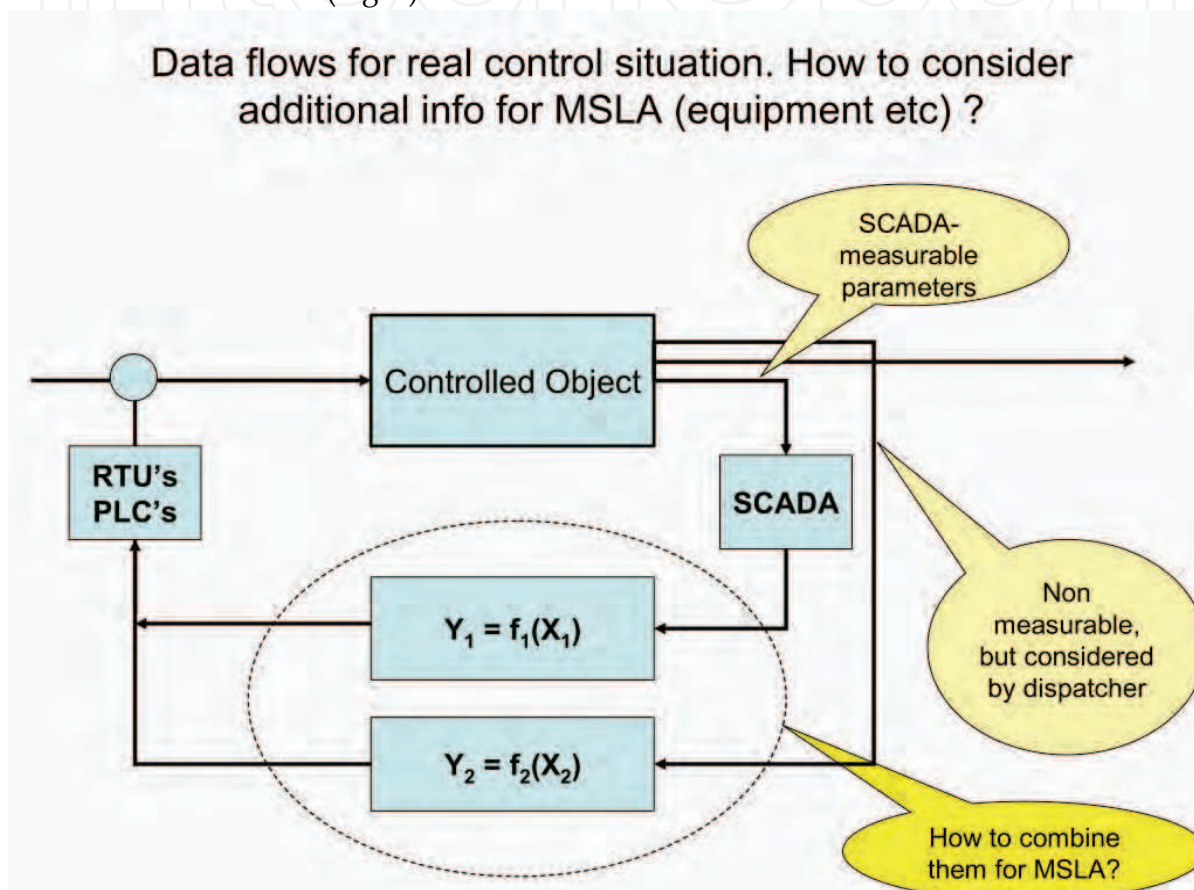


Fig. 2. The real control situation

We can see that operational staff (dispatcher) works in other reality and manages necessary types and sources for additional data, every time may be with other method. The real situation of control with using of MSLA is:

- Reading a part of data using SCADA;
- Based on this input, the system determines what additional data is required in order to correctly modify the output action, and where this data is located;
- Definition types of additional data, sources of them and extraction methods;
- Searching and receiving these additional data;
- Combining all data from all sources, analyze;
- And, finally, defining the appropriate control action based on all the data received from all sources.

The mentioned practical features are light version these things, which pull and feed the control theory for flow technology today. They determine a way which must be passed (done) and tasks which must be solved.

3. Discussion about concept of Technological Coalitions

Before we start to consider the above mentioned different external changes and try to adapt MSLA for them we should spend some time for decomposition of technological processes and to give some necessary previews and introductions for new ideas. First of all we are going to introduce and discuss a new concept named "The Technological Coalition" as a special part of technological process and corresponding part of algorithms for its control.

We use decomposition of technological process not as it is used in most cases, not only as a tool for decreasing the complexity of technology description. We use it as tool for finding and delimiting areas of instability, changeability in the technological process. What does it mean? It means that any new change (factor) which will appear will be localized in Technological Coalition (TC).

The second assumption is that all TCs behave in the same way. The same behavior means that the same operations can be determined.

$$TC = \langle A, R, LCA, M\Phi, M\Psi, MS, LC \rangle$$

A - set of separate devices for different technological needs. Any reservoirs, valves, sand collectors, pumps, separators, drip pockets, desalters (demineralizers), heaters, freezers, fractionators, precipitation tanks, coolers, sewage tanks, components for flare system, boosters and etc. They are collectively known as "equipment" or "devices" or "aggregates". Each type of equipment (aggregate) has its own local control algorithm (included in a special set named LCA - see below).

R - defines physical links which connect product inputs and outputs of different aggregates. It means for most cases some pipes or other transporters. The flows of different substances pass through them and various parameters of these substances change in the process.

A and R together build a TN - an oriented graph of Technological Net. The concept of a TN is too well known to require any examples.

LCA - set of local control algorithms for each type from A. We prefer to use Moore-Automat Model for each element of LCA, but it isn't necessary.

$M\Phi$, $M\Psi$, MS are tables with a special purpose - they collect changes and give possibility to consider and process them.

LC - Life Cycle of TC (see below) described as an oriented graph having six special states.

The traditional division into the controlled object (technological process) and the control system (algorithms, MSLA) is changed here. Please note that A, R, LCA represent the flow technology, LC - is a part of control system. The TC consolidates parts of both sides. And the TC isn't the result of decomposition of our flow technology only.

The TC as an abstract idea doesn't have exactly one, precise and absolutely clear interpretation for operating staff and for software developers. We realize it. In most cases we can associate the TC with an idea of "route" (as sequence of technological devices), but not always. Note that a list of TC's appears on the phase of pre-design of control system, but implementation is often not clear on this phase. Moreover, it will be better when technological specialists and control specialists develop the list of TC's together.

It is often right that all devices of TC serve different characteristics of whole stream of substance. This stream has technological comprehension as indivisible (is viewed as indivisible). On the other hand, we suggest that MSLA should only be used to control such controlled object as TCs, not for other purposes. Thereby we'll define the special rule of correct using for term "MSLA". But a problem of coordination between different MSLA will appear. We'll try to find approach for decision of it later. So we have some important assumptions now:

- TC is possible. In other words, we can show that any single change which appears in the flow technological process or in equipment after retuning (replacment) will upset not all technological process but only one delimitable part of it. We'll call this part of whole technological process as TC but we'll understand it as special combination of controlled object and controlling object having formal definition.
- There is the common control architecture for all TC's irrelative of their size and local behavior. (Really it is only hypothesis. We are going to prove it later.)
- An operator will be able to manage TC using special tools.

Since we have defined TC as the controlled object, we have to explain what are the control commands for it. We assume that we do not have to physically build the TN, so our commands will not create structures, but deal with different states of a TC with an existing structure. Our commands are not commands for structure building. First of all we have to determine the needed states of TC and after that we'll determine appropriate commands. The commands will tell the TC to change from one state to another. Simplest way is to determine two states (is working now and isn't working now) but this will not be of much practical use. This is a very general view, of course. It contradicts none of the existing views but is not yet much help otherwise.

We need of course more pragmatic content for TC. How and where can we get meaningful states for TC? We shouldn't forget that we are going to invent and to use decomposition idea which allows to have the same behavior for all TCs. We should analyze the technological reality again. We can imagine and correct understand these operations as wake up, prepare, launch, get current state, tune, customize, shut down (and probably more additional commands) without semantic troubles. So we want to introduce some overview of possible states of TC's for future control systems designers. These commands are meaningful for any TC and they determine moving between states at the same time. Involving and shutting down of TC's won't be momentary. There are often a multi-step involving (preparing) process and/or multi-step shutting down (canceling) process. So, states of TC we can see on Fig. 3. By the way - there is a right question why number of states is equal to six? Is it a special requirement of flow technology or not? Are there any exceptions? Our answer - the real number of TC's-states may be more or less than six, of course. It depends on concrete application. Important thing is the equal number of states for all TC's. Only if all TCs have the same states in their LC we can suggest the universal control mechanism for control of them. But for flow technological processes six as number of states is preferably.

By the way, all these commands will be realized as methods for object oriented concept of TC's if somebody is going to put much effort in OOP-implementation of TC's.

Moving (by operator's supervision or under control of the automation system) through these states is Life Cycle (LC) of any TC. We have steps for operator and some other steps for automated control system now. A clear division between an operator and automated control

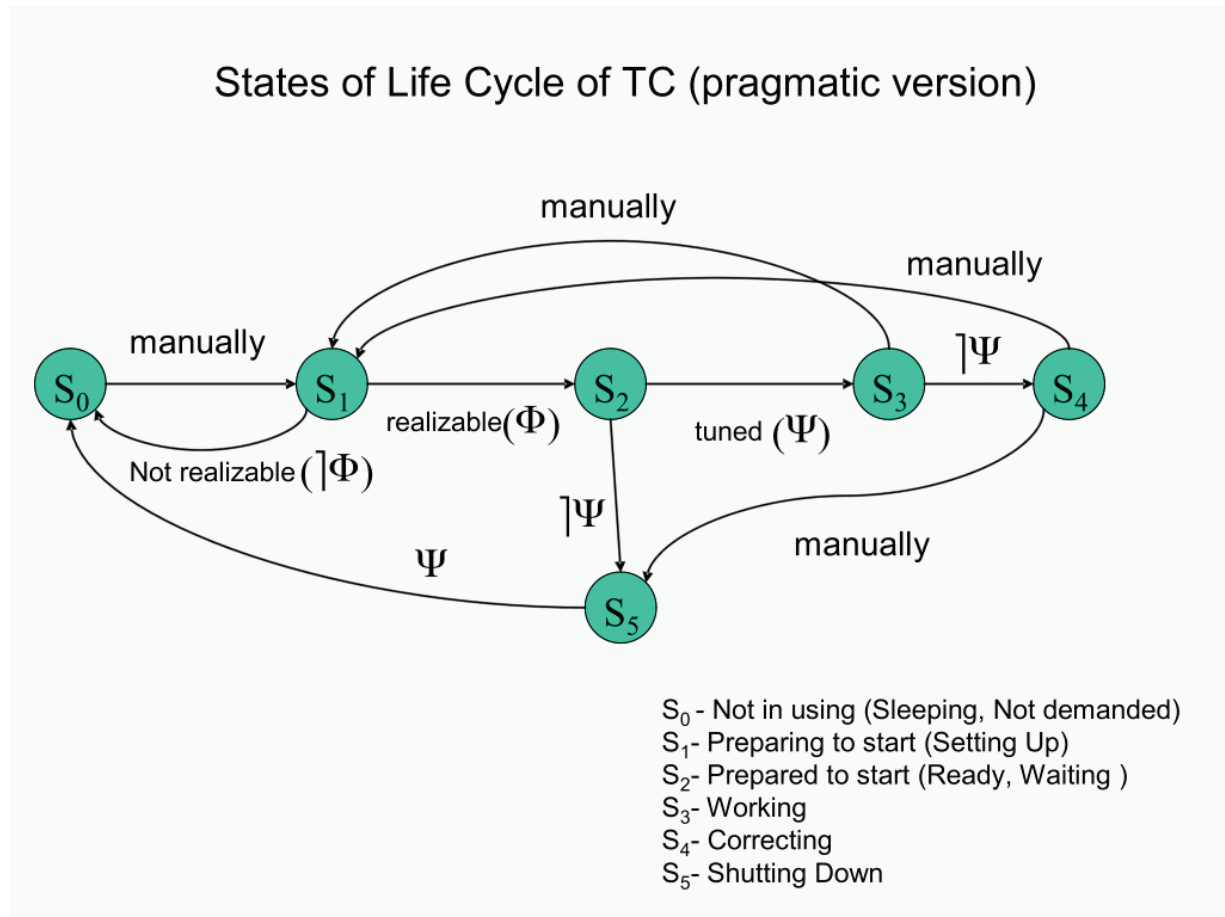


Fig. 3. The possibly states of LC

system allows to divide future efforts. Transitions marked “manually” need only right-designed human-oriented interface. As we can see transition marked otherwise need to connect with sensors and/or SCADA. There are some comments to transitions:

- $S_0 \rightarrow S_1$: First transition after sleeping. This transition managed by operator manually. Reasons for activity of dispatcher in this transition are out of this paper. Dispatcher can reject from his decision about waking up if it will necessary.
- $S_1 \rightarrow S_2$: Preparing to start (phase one). Intensive using of **MΦ-table** (see below). Operator fills in this table self or asks technologist. Meaning of this step - to collect all necessary devices and to check them (they are in good working condition) and avoid involving of them in other active TC's. If realizing =OK then jump to S_2 , else jump to S_0 and sending message to operator. If we have conflict(s) (necessary devices isn't free or not ready) then dispatcher can launch a special local subprocess for this aggregate.
- $S_2 \rightarrow S_3$: Preparing to start (phase two). Intensive using of **MΨ-table** (see below). All necessary devices are included in TC but are not ready to work yet. For correct launching we must to prepare additional conditions. Level in tank_2 must be ≥ 3 m, for example. Or temperature of oil in pump must be $\geq 50^\circ$ C for correct starting, etc. There conditions can have logical or discrete or analog values. We associate them with devices (aggregates). The common conditions can exist too, certainly. Operator must launch and finish some additional local subprocesses for each device if it is necessary (oil-heating in bearings of involved pumps or filling of tank to necessary level, for

example). As result of this step we have a set of sequences for launching main technological process associated with TC. For example (abstractly): If $(Level_{12} > 3)$ then A4 (open). When all launching commands executed then the state of TC switches from S_2 to S_3 .

- $S_3 \rightarrow S_4, S_4 \rightarrow S_1$: While we have S_3 the technological process is working normally. This is area for 1st and 2nd types of algorithms. Operator can solve to use slightly different configuration of technological devices. But operator doesn't want to use another TC. For example he (she) wants to start only an additional pump. Probably it is temporary changes. Anyway, it is necessary to check information about additional technological devices: jump to S_1 . After checking (if "true") we return through S_2 to S_3 .
- $S_3 \rightarrow S_4, S_4 \rightarrow S_5$: Operator have solved to change TC. Preparing to shutting down, checking for special conditions is needed. Operator usually has to use special commands or local procedures (manually or automatically). Changing of states $S_4 \rightarrow S_5$ means that all conditions are "true" and we can start shutting-down procedures immediately when we want.
- $S_5 \rightarrow S_0$: Shutting down procedures are finished. Shut down of TC is complete.

Most likely that S_3 is the state in which TC stays maximum period of time. It is normal but we shouldn't forget about other states. It is well known that for example an airplane has normal state (the flight) maximum period of time but the more dangerous and more required for the precise control are the other states (take-off and landing).

It is clear from practical experience that some devices for technological reasons can sometimes change their belonging to TC. It is true but each device must belong to only one TC at any given moment. In our oil processing example we stated that raw oil from different oil fields contains slightly different levels of sulphur. It requires different equipment and different routes (different connections) for processing. So, the staff should switch some pipes, pumps, valves which are serving other routes now. It means that our opinion about temporary belonging to TC is mainly true for pipes, pumps, valves. There is a special state S_4 in which it is possible. If TC has received external request for some device then there are some different variants of TC-reactions in this situation. For example:

- Check current availability of device. If it is free now then just "to lend" it
- If there is not availability then to ignore external request
- "To lend" required device to another TC but after finish shutting down procedure for current (giving) TC (postponed lending) but to start shutting down procedure for current TC
- Other scenarios...

Please note the following. On the one hand, we localized correct area for MSLA using (only for TC). On the other hand, we declared standartized LC for TC. From this it follows that MSLA can have standartized structure. In other words, we can build one algorithm for any TC if only each TC will have the same LC. In that way we changed an old approach. We suggest to modify MSLA's changes considering practice from building a new algorithm every time if only we fixed some changes to configuring one time developed algorithm. It is important thing. MSLA will be standartized part of conrtol system now.

It is clear that MSLA's aging problem didn't disappear with suggestion of TC. We could only localize external influences without considering them. We also need a special generating tool which must be available for using not in design phase but in running phase (see Fig. 4). Probably it will a special extension of SCADA-software.

2-step changing of MSLA by using new data

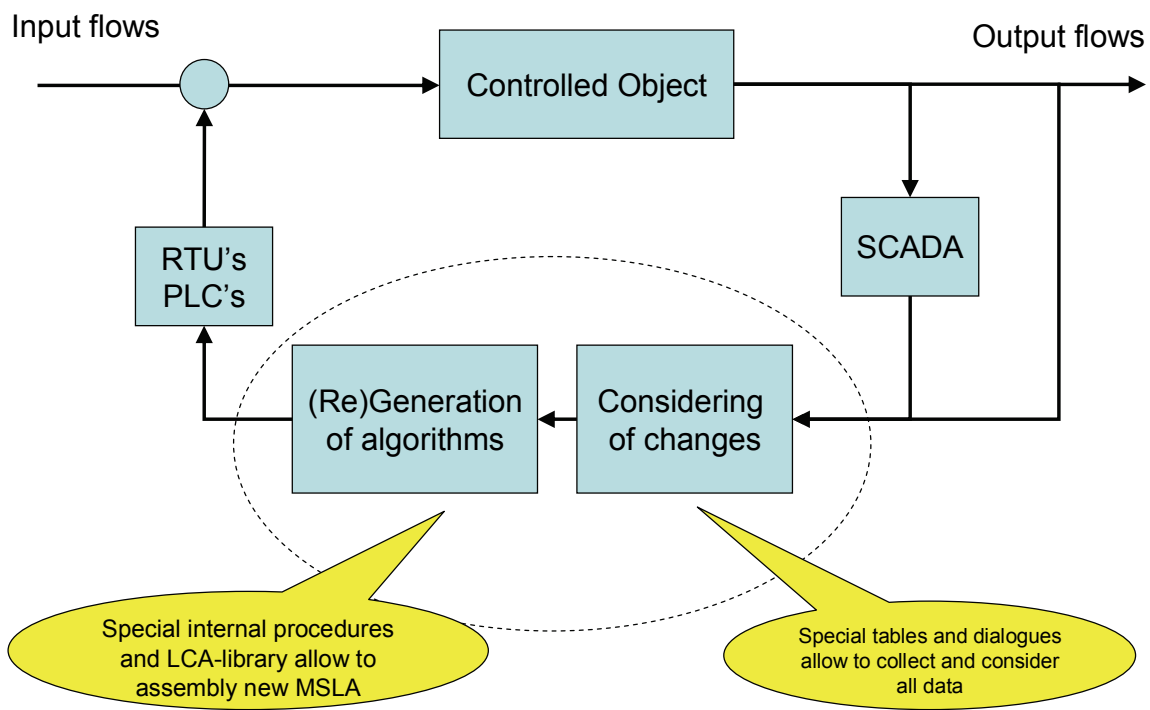


Fig. 4. Including the considering and generating parts in the feedback loop

4. Tools for external changes management

If we return to TC's definition then we can see there some MS, $M\Psi$, $M\Phi$. Yes, there are some tables which describe all involving aspects for each device. The horizontal axis is devices from A, vertical axis is set of foredesigned TC's.

The first table is MS. It contains device's states needed to involving to any TC, states for starting of any TC. It is clear that different TC's can theoretically require different starting states from the devices. All states for all devices we can get from Local Cycle of Aggregate (LCA). Each LCA is a simple FSM for one device. We can suppose that LCA is a part of TC. Or, otherwise we can think that LCA is a common information resource (like a software library), external for all TC's. Important that we can extract from LCA command sequences needed for transition from any state of given device to any other state.

If we have current states (we will use an additional table MT for current states of technological devices - from SCADA) and states from MS it seems after that that we'll be able to assembly TC launching program only with conjunction different command sequences for any device. We think it will be better when we postpone mentioned assembling yet. Now it is the best moment to consider last changes which we discussed formerly. We are going to suggest using two new tables $M\Psi$ and $M\Phi$. All additional conditions which must be considered are entered into these tables. Commands which are prepared from LCA must be sent to controllers after allowing conditions from $M\Psi$ and $M\Phi$.

Example of $M\Phi$ -table – conditions for involving aggregates in each TC (readiness and not in repairing)

	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉
TC ₁	H ₂ =1	q ₁	B ₉ <2	F ₇ =4	q ₃		q ₄		q ₅
TC ₂		q ₂	q ₂	F ₂₂ =1	K ₂ =1	q ₃	U ₁₂ <28	q ₁	
TC ₃	q ₃		(U-Y)<4	q ₂	q ₃	K ₁ =1	q ₁	Q ₅	KL ₁₇ <5
TC ₄	K ₁ =1	(U-Y)<5		F ₂₃ =1	K ₁ =1		Q ₂₃ <90	KL ₂₃ <0	
TC ₅		(U+K) ₂ > (K+H)		q ₄	q ₄	F=44	KL=<9		q ₇

Example of $M\Psi$ -table – additional conditions for involving aggregates in TC

	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉
TC ₁	H ₂ =1	q ₁	B ₉ <2		q ₃		q ₄		q ₅
TC ₂		q ₂	q ₂		K ₂ =6	q ₃	U ₁₂ <28	q ₁	
TC ₃	q ₃		(U-Y)<4	q ₂	q ₃		q ₁	Q ₅	KL ₁₇ <5
TC ₄		(U-Y)<5			K ₆ =1		Q ₂₃ <90	KL ₂₃ <0	
TC ₅		(U+K ₂)> (K ₂ +H ₂)		q ₄	q ₄	F=44	KL=<9		q ₇

5. General mechanism of considering and control

TC is functioning not alone. There are some other TCs, which can at the same time launching, working, configuring, shutting down. The right environment for the one TC are the other TCs.

There are two virtual sets in our vision: a Set of Active TC's (SAC) and a Set of Passive TC's (SPC). In a real production process each TC belongs to SAC or to SPC. The changing between SAC and SPC under supervision of dispatcher or under special algorithms is the abstract vision of our flow technological process. Objects for changing between SAC and

SPC are the TC's (see Fig. 5). Let us agree that integrated flow technological process for each moment of time is the SAC. Any of TC can change its current belonging (to SAC or to SPC) during technological process a lot of times. It depends only on technological needs and/or dispatcher's will (wish).

Destination of the control system in this vision is supporting correct changing (TC-moving) between SAC and SPC according to technological needs and operator's will. Inside this task there is another task, more local, but no more important: to support the LC of each TC.

The general vision of process control with using TC's

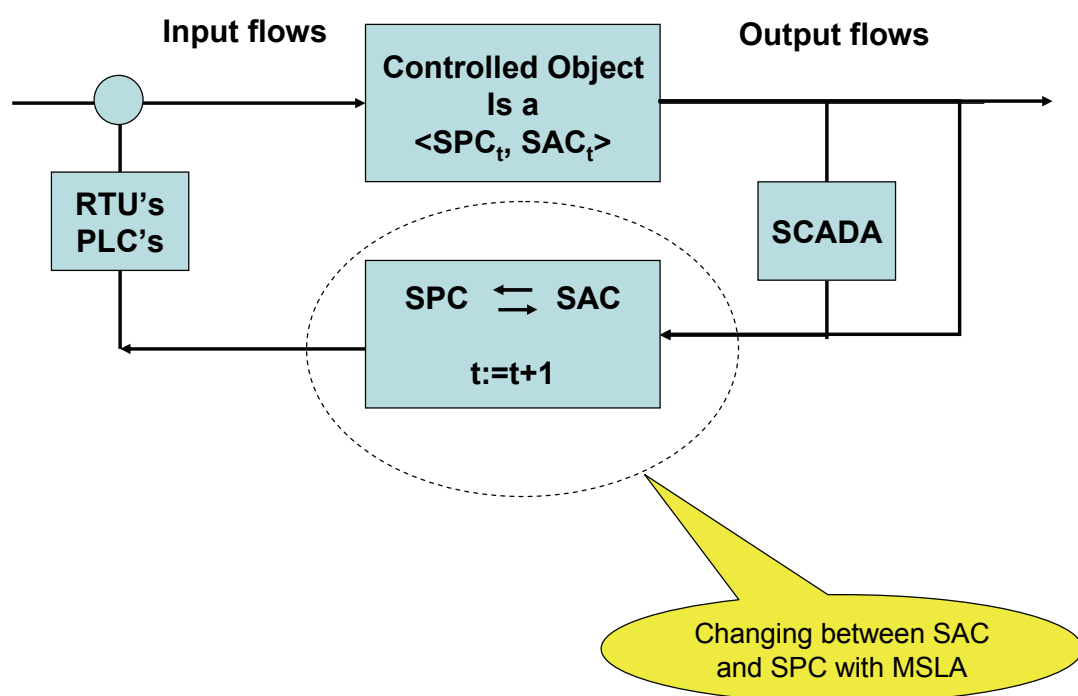


Fig. 5. SAC and SPC are the main controlling parts.

When we have certain SPC/SAC and want to change SPC/SAC for next point of time we'll do the same actions for any points of time. These actions are included in MSLA. Note that the MSLA is not any multistep algorithm. It is the multistep algorithm having TCs as controlled objects and working with SPC/SAC. It is possible to have a lot of working instances of MSLA: each one for serving one TC (its LC). Steps for any MSLA and for any states of LC are equally.

How does it work together? The behavior and steps of high-level interpretation mechanism for MSLA are the following:

- All TC's belong to SAC or SPC. All TC's including in SAC are working. Low level automated control systems (PLC's and RTU's) are working, structure of flows is defined by an active TC's, flows function are under control of alarms and local regulators, and a set of actual events is formed.

- Operator can observe active TC's (using SCADA) and can understand if they are working correctly.
- Depending on the real situation in manufacturing, operator selects a necessary strategy by launching and shutting down for each TC. Once time operator makes decision to change SPC/SAC: to launch a TC_j or to shut down TC_k (some external events have occurred). Operator selects a concrete TC to launch or to shut down manually and after that he (she) can entrust the matter of control to MSLA (MSLA begins to implement control mission). Current states of all needed devices are read through SCADA (by MT-table). Possible collisions (sharing some aggregates with another working TC's) are solved by operator using special human-oriented dialog.
- Preparing to assembling starts when all collisions are solved. If necessary the monitor (or operator) makes some queries to fill in the special tables for actual data (new conditions for involving devices are possible). $M\Phi$ and $M\Psi$ are using now. The monitor reads a new data from mentioned tables. Low level vision of MSLA for PLC-executing is set of sequences "condition→action". Two parts of data are combined by logical assembling in the one multi-step program. This set of sequences is goal of assembling and it requires two types of source data - new conditions (from $M\Phi$ and $M\Psi$) and new actions (from LCA)
- Assembling of programs starts. Monitor reads current and targeted states. If LC-graph has transition with $M\Psi$ or $M\Phi$ for these states then monitor makes data reading. Most important by launching is transition from S_2 to S_3 (see LC-graph of TC) and by shutting down - transition from S_4 to S_5 . By generating of control a special logical assembler (SLA) extracts sequences of necessary commands from the mentioned LCA-library. By generating "shutting down"-program the SLA uses the LCA too. Logical assembling is completed when we have the list of instructions (abstractly example): if (conditions from $M\Phi_i$ and $M\Psi_i$ are "true") then `extract_commands_from_LCAi(MT_i , MS_i)`. A number of sequences equal of number of devices. Mentioned in expression above substring "`extract_commands_from_LCAi(MT_i , MS_i)`" means that the SLA expands this command (as whole instruction) into set of commands based on the accordingly finite automat from LCA. It is important to note that the SLA makes only substitution from the LCA for each instruction. The necessary order (sequence) of turning on of different devices in the real flow we can get by using **$M\Psi$ -table**. For example we can add to formal conditions for aggregate in **$M\Psi$ -table** a special conjunctive term for considering that previous device got right state before.
- Finally, the algorithm for launching TC_j and (or) "shutting down" TC_k is assembled and ready to start now. The monitor or operator launches each assembled and ready to start "fresh" algorithm. Local PLC's and RTU's must implement this algorithm after loading instructions. Special software for uploading a programs into memory of PLC's is available and we don't focused on it here.
- Launching and shutting down processes are working and controlled by operator. Monitor receives back answers from PLC's and RTU's.
- If processes have finished OK then would be to refresh (to update) SAC/SPC. MSLA is complete. Go to 1.

Note, we didn't formalize merging and dividing of different TC but it is possible in nearest modifications of the control mechanism. The special mechanism for sharing (or for "lending") several supporting devices (mainly such as pumps) between different TC will be

described in next publications of authors. So we have that slightly corrected principle of decomposition (we are looking for and use coalitions of technological devices which have standardized behavior - LC) and not complicated extracting- and re-assembling procedures allow to have standardized MSLA as part of control system and to get rid of mentioned problem of "aging". The general view is on the Fig 6.

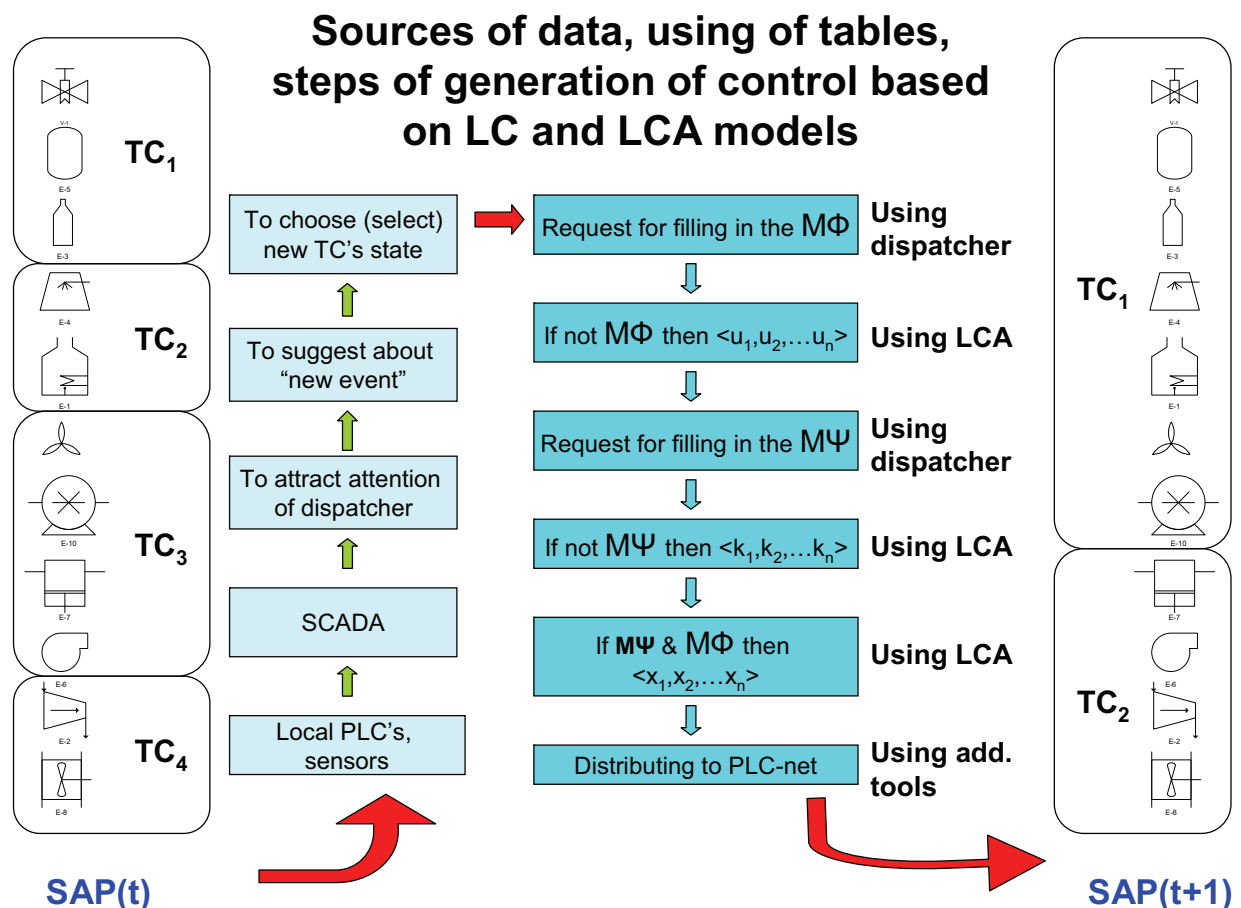


Fig. 6. All components are working together.

6. Conclusion

It was stated earlier that of the three types of control which were analyzed the MSLAs are the most likely to get out of date. Moreover, in most practical cases MSLAs work best immediately after being first implemented and started up, after which error accumulation inevitably begins. It is not a good idea to become reconciled to this fact. We have realized that classical FSM-approach doesn't work in practical cases of control. It causes MSLAs to fall into disuse, but current disadvantages of MSLAs are not intrinsically insuperable. In any case it is now unacceptable to go from automation back to manual control. Today's industries require more and more automation for increasingly complex technological processes. But as of today the real technological equipment is not yet like P'nP devices and not all necessary control standards are implemented or even exist. We hope that we were able to explain why the classical FSM approach leads to increasingly unsatisfactory performance of MSLAs in real life situations. Their developers didn't consider possible

changes in control logic after maintenance, repair or technological changes. This destroys MSLA in the end.

We need to return to the reality of big plant control. FSM is able only to transform strings $\alpha \rightarrow \beta$ but real control has more than one step. The real control situation must assume the worst thing: that the controlled object has changed. On receiving information from the controlled object there is often a choice (or alternative) $\alpha \rightarrow \beta$ or $\alpha \rightarrow \gamma$ and we need additional information to make the right choice. The real situation is “if (α and Ψ) then β else γ ”. Ψ is that additional, often even non-formalized, but technologically meaningful information, not received from SCADA usually. It is important to make the transition from the fully determined situation of string transformations to the real situation of big plant control. Note, that type 2 algorithms (PI, PID) are inherently adaptable (since coefficients can be tweaked) and are in the control situation from the beginning, but MSLAs are not.

How to impart such adaptive potential to MSLAs, which are rigid and inflexible by definition? We can try and anticipate all possible changes in our system and represent them as distinct states of the FSM. However, the total number of such states will soon grow so huge that we will not be able to perform the necessary calculations. We know that we’ll bump into the dimension problem. This proves that this is the wrong way. But as technological changes are unavoidable and cannot be ignored, they must be classified and considered. The right (new) way is as follows. We introduce into the feedback loop our model with TC’s states and MS, $M\psi$, $M\Phi$. Our approach allows to:

- Identify the current state of the process in the controlled object.
- Understand which information must be gathered additionally for this particular state.
- Generate the correct control incorporating the additional information during assembling procedure.

The classical FSM performs only 1st and 3rd tasks. Moreover, the FSM performs 3rd task with a one-step fully predefined function. We implement this task with a special command-generating procedure.

So, after the identification of the current state by means of our model (incorporated into the feedback loop) we suggest that outputs should not be generated right away, but with a delay for gathering the additional information (MS, $M\psi$, $M\Phi$) and assembling controlling outputs using LCA. Now we can point out exactly where the adaptive potential of MSLAs is. It appears only if we change single-step FSM functions to two-step procedures.

First, we introduced the concept of TC. The initial conception, building, implementing of any TC must be realized very carefully and with full attention to details. We are sure that only cooperation between technologically thinking people and experts in the area of control systems can give useful results, at least in the first stages. After that we’ll have some experience and will be able to construct any TCs correctly. TC can help to solve problems caused by huge unwieldy MSLAs and can localize (and subsequently process) external changes.

A word or two about other possible uses of our approach. For example, we know that there is a problem for driverless (fully automatic) cars to drive from point A to point B in the city. Moving through city, from one intersection to the next intersection is essentially like MSLA. Crossroads are points for collecting new information (new changes) and generating new control output. TC is a part of route in which appeared new information doesn’t affect to decision making and routing.

To sum up, we can hope that some principles which allow to build the new control system for the flow industries have been here developed and explained. The new control system has adaptive potential which helps to cut down maintenance costs.

7. References

- Akesson, K., Flordal, H., Fabian, M. (2002). Exploiting modularity for synthesis and verification of supervisors. *Proceedings of the IFAC World Congress*.
- Ambartsumian A. A., Kazanskiy D.L. (2001) *Technological process control based on event modelling. Part I and II*. Automation and Remote Control, №10, 11; 2001.
- Ambartsumian A. A., Kazanskiy D.L.(2008) The approach of complex technology automation with using of discrete event models in a feedback control , *Proceedings of 17th IFAC World Congress, Seoul, 2008*
- Cassandras, C. G., Lafortune, S. (2008) *Introduction to discrete event systems*. Dordrecht: Kluwer Academic Publishers, p. 848.
- Golaszewski, C. H., Ramadge, P. J. (1987). Control of discrete event processes with forced events. *Proceedings of the 28th Conference on Decision and Control*, pp. 247–251, Los Angeles.
- Gaudin, B., Marchand, H. (2003). Modular supervisory control of asynchronous and hierarchical finite state machines. *In European Control Conference*, Cambridge.
- De Queiroz, M. H., Cury, J. E. R. (2000). Modular supervisory control of large scale discrete event systems. *Discrete Event Systems: Analysis and Control, Proceedings. WODES'00*, pp. 103-110.
- F. Zambonelli, N. Jennings, M. Wooldridge. (1994) Organizational rules as an abstraction for the analysis and design of multiagents systems. *International Journal of Software Engineering and Knowledge Engineering* (1994).
- Edgar Chacon, Isabel Besembel, Jean Claude Henet. (2004) Coordination and optimization in oil and gas production complexes. *Computers in Industry №53; 2004*. pp. 17–37.
- N. Jennings, P. Faratin, A. Lomuscio, S. Parsons, C. Sierra, M. Wooldridge. (2001) Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10 (2), 2001, pp. 199-215.
- Wonham, W. M., Ramadge, J. G. (1988). Modular supervisory control of discrete event systems. *Math Control Signals and Systems*, 1, pp.13-30.
- Yoo, T.-S., Lafortune, S. (2002). A general architecture for decentralized supervisory control of discrete event systems. *Discrete Event Dynamic Systems: Theory&Applications*, 12(3), pp. 335-377.



Process Management

Edited by Maria Pomffyova

ISBN 978-953-307-085-8

Hard cover, 338 pages

Publisher InTech

Published online 01, April, 2010

Published in print edition April, 2010

The content of the book has been structured into four technical research sections with total of 18 chapters written by well recognized researchers worldwide. These sections are: 1. process and performance management and their measurement methods, 2. management of manufacturing processes with the aim to be quickly adaptable after real situation demands and their control, 3. quality management information and communication systems, their integration and risk management, 4. management processes of healthcare and water, construction and demolition waste problems and integration of environmental processes into management decisions.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Alexander Ambartsumian and Dmitry Kazansky (2010). An Approach to Technological Processes Automation using Technological Coalitions Based on Discrete Event Models, Process Management, Maria Pomffyova (Ed.), ISBN: 978-953-307-085-8, InTech, Available from: <http://www.intechopen.com/books/process-management/an-approach-to-technological-processes-automation-using-technological-coalitions-based-on-discrete-e>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen