The contour of causality in control automata of systems

Mykhailo Poliakov[a], Sergey Subbotin[a] and Oleksii Poliakov[b]
[a] *Zaporizhzhia Polytechnic National University, vul. Zhukovs'koho, 64, Zaporizhzhia, 69063, Ukraine*
[b] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", prosp. Peremohy, 37, Kyiv, 03056, Ukraine*

Abstract

Known models of control automata of systems describe only the necessary cause-and-effect relationships between sets of an automaton, except for the relationship between sets of outputs and inputs. This complicates the analysis of non-standard situations associated with the diagnosis and development of machines. A model of an automaton in the form of activity and control loops is proposed, which reflects the cause-and-effect relationships in the control automata of the system, including the relations between the elements of the set of outputs and the set of inputs of the automaton. These circuits interact through the machine states.

The classification and models of the machine states are proposed. The structure of the knowledge base along the automaton contours and the predicates types describing causal relationships in the automaton are proposed. An example of constructing the contours of activity, control and Prolog program for diagnosing the technical state of the machine using the knowledge base is given.

Keywords 1

Control system, control automata, causal relationships, contour of activity and control, knowledge base.

1. Introduction and Analysis of Publications

In systems for technical purposes, it is customary to distinguish an object and a control unit [1]. The system behavior is determined by the structure of the functional converter [2], acting as control automata, as well as by the nomenclature and characteristics of the operating automata of the control unit.

The control machine is described by a logical and semantic model. The logical model of control automata is a FSM, which is described by a tuple [1]

$$A = <X, Y, S, s_0, \delta, \psi>, \tag{1}$$

where X is a set of inputs; Y is is a set of outputs; S is a set of states; s_0 is an initial state; δ, ψ is a function of outputs and transitions of the finite automaton, respectively.

The logical model of the control automaton arose as a result of the formalization of the verbal description of the system's behavior in the form of a set of statements [3]. The use of this model in the construction of control systems has shown its limitations, which lies in the fact that it takes into account only the predetermined behavior of the control object and does not take into account the ring causal relationships in the contours "control object – control unit" in case of unplanned changes in the control object, the external environment and the technical condition of the elements of the control unit.

In [4], the influence of the outputs of the control automaton on its inputs is modeled by constructing a control loop, which includes the control automaton and the finite automaton representing the control object. This circuit is used to synthesize a control automaton, which specifies

---

CMIS-2021: The Fourth International Workshop on Computer Modeling and Intelligent Systems, April 27, 2021, Zaporizhzhia, Ukraine
EMAIL: polyakov@zntu.edu.ua (M. Poliakov); subbotin@zntu.edu.ua (S. Subbotin); poliakovalyosha@gmail.com (O. Poliakov);
ORCID: 0000-0002-7772-3122 (M. Poliakov); 0000-0001-5814-8268 (S. Subbotin); 0000-0002-9355-7056 (O. Poliakov)

CEUR Workshop Proceedings (CEUR-WS.org)

a certain behavior of the system object. At the same time, the work [4] does not detail the structure of the control loop, does not consider objects with continuous behavior, as well as cause-and-effect relationships that arise in the event of a change in the technical state of system elements.

Further development of logical programming methods and the growth of requirements for adaptation and development of system behavior led to the use of semantic information to describe the behavior of the control automata.

The sematic model of control automata element is a system of tuples [5]

$$SMS=<S_a, S_c>; S_i=<N_i, CD_i, KB_i>; N_{ij}=<N_p, N_{pr}, N_f>; \quad (2)$$

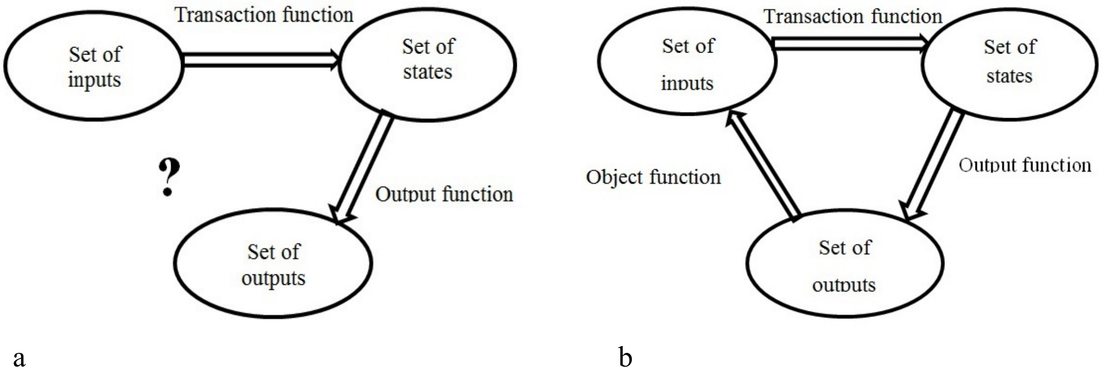
where S_a, S_c is a set of activity and control contours, respectively; S_i is a set included in the sets S_a, or S_c; N_{ij} is a complex of names of the j-th element in the i-th contour; CD_i is direction of causality in the i-th contour; KB_i is knowledge base of the i-th contour; N_p, N_{pr}, N_f, are the set of names of the j-th element in the i-th contour in the style of the prehistory, the current choice and the expected events, respectively. The knowledge base consists of rules of the form KB_{ij}: IF CD_i THEN $N_{ik} <= N_{im}$, where KB_{ij} is an element of the knowledge base of the j-th element in the i-th contour. The expression $N_{ik} <= N_{im}$ indicates the fact that N_{im} is a consequence of N_{ik}, where N_{ik}, N_{im} are knowledge in the form of names for elements k and m in the i-th contour. At the same time, work [5] does not detail the role of states of an automaton in the interaction of activity and control circuits, does not describe a method for constructing cause-and-effect relationships in hierarchical states.

2. Research Objectives

The purpose of this article is to provide research on aspects of the behavior of systems associated with a change in the technical state of their elements and parameters of the control object by expanding the knowledge base of the system, detailing the interaction of the activity contours and control of the system through the states of the automaton.

3. Research Content

As you know, the "classic automaton" has three sets, which are connected by two functional relationships, as shown in Figure 1a.



a b

Figure 1: Functional relations in the "classic" (a) and the proposed (b) automaton

In this version of the description of the automaton, and its semantic version, there are causal relationships between states and outputs (actions), as well as between current states, inputs and subsequent states. But in this description there are no causal links between outputs (actions) and inputs (events). In addition to the control machine, the control object and the operating machines of the control device take part in the formation of these connections.

The proposed function of the object (Figure 1b) establishes the functional causal dependence of the inputs of the control machine on its outputs. In a more detailed description of the function of the object, the functions of influences and reactions should be taken into account, as shown in Figure 2.

The elements of the "control automaton – object of control" system form the contours of the activity and control of the finite automaton, which interact in the states of the automaton. Let us clarify that the activity contour describes a chain of cause-and-effect relationships ensuring the continuation of the impact on the system object in a given active state. A control loop describes the cause-and-effect relationship that controls the transition of the automaton to another state. Accordingly, the number of control loops passing through this state is equal to the number of transitions from this state to others.

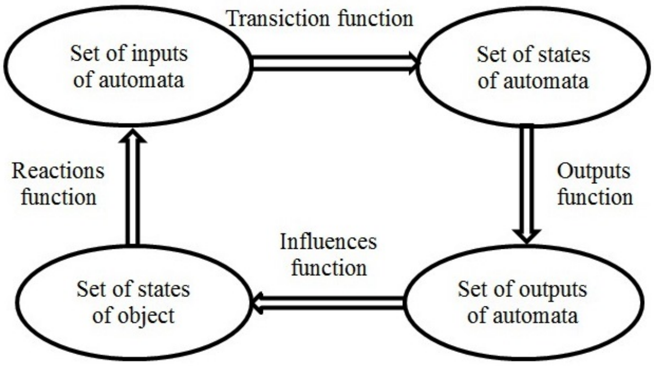


Figure 2: Functional relationships in the control machine detailing the functions of the object

To describe the relationship of the state with the rest of the machine, we introduce the following elements: event control inputs (ECI) – inputs) and event control outputs (ECO) of control loop events, activity outputs (AO) – and reaction of object inputs (RI) for these actions in the contours of activity [5]. Between these elements of the external interface of the state in the framework of Laplace causality and the direction of causality from entry to exit of the state, the following types of causal relations are possible: ECI <= AO, (activity is a consequence of the state transaction); RI <= ECO (transaction is a consequence of the reaction).

If in some state of the machine intersect certain circuits of activity and control, then the above causal links provide the interaction of these circuits as shown in Figure 3.

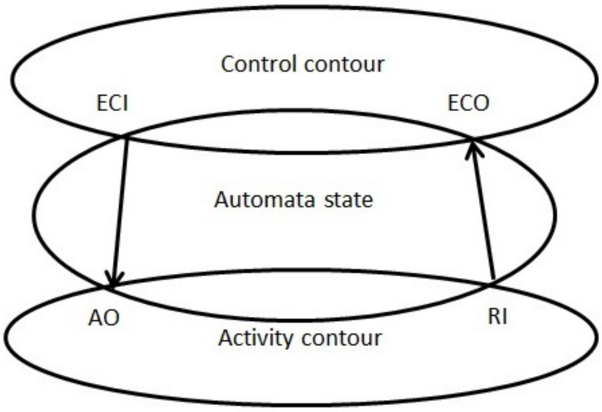


Figure 3: The interaction of the contour of activity and control in a certain state of the machine

The mechanism of interaction is as follows:

1. The change in the circuit of activity comes to the input of the reactions of the RI state.
2. Due to internal causation in the state, it is transmitted to the ECO output and is the cause of changes in the control loop.
3. Changes in the control loop lead to activation of the ECI input.

4. Due to the internal causal links in the state, Due to the internal causal links in the state, this activation is manifested at the output AO and initiates changes in the contour of activity.

A special case of the control circuit is the arc of the graph of the automaton, which exits state A and closes in the same state.

With the help of causation and the external arc of the graph of the automaton, which comes out of state A and closes in the same state, you can perform a transition forward (conditional causality) or backward (looping of the site) in the control or activity contour.

To enter the cause-and-effect relationship ECI1 <= ECO2 (transaction2 is a consequence of transaction1 or the original transaction event is a consequence of the input transaction event) in state A we use the external arc of the automaton graph, which exits AO of state A and closes to RI input in the same state and the internal connection RI <= ECO. Similarly, using an external arc between the ECO output and the ECI input, you can create a ratio of RI <= AO (activity is a consequence of the reaction).

The proposed classification of states of the control automaton depending on the logic of processing input events is given in Table 1.

Table 1

Classification of states of the control machine

Type of condition	Classification feature	
	Incoming event processing	Memory of previous events
Switching	is absent	is absent
Combinational	is present	is absent
Automatic	does not affect	is present

In the "classic" finite state machine, the output (activity) does not depend on how the activation of the state is performed. That is, if such a state has more than one input event, then it is a combinational state with the processing of input events by logic OR in relation to the activity output of the state. And the automatic state in such automatic machines represents the control subautomata unit of the lower level. This machine is executed if the state in which it is placed is currently active.

The state of the S_i machine can be described by a tuple

$$S_i = <\boldsymbol{ECI}, \boldsymbol{RI}, \boldsymbol{ECO}, \boldsymbol{AO}, \mu_s, t_s>, \tag{3}$$

where ***ECI*** is a set of inputs of control events; ***RI*** is a set of inputs of reactions to activities; ***ECO*** is a set of outputs of control events; ***AO*** is a set of outputs of activity events; μ_s, t_s is a function of activity and state transactions, respectively.

The μ_s function describes the mapping of the set of ***ECI*** inputs to the set of ***AO*** outputs μ_s: $\boldsymbol{ECI} \times \boldsymbol{AO}$, and the t_s function describes the mapping of the set of ***RI*** inputs to the set of ***ECO*** outputs t_s: $\boldsymbol{RI} \times \boldsymbol{ECO}$.

The type of functions μ_s, t_s depends on the type of state. For the switching state it is enough to list the relations ECI_i <= AO_j and RI_i <= ECO_j, where i, j are the numbers of elements in the corresponding sets.

To describe the combinational state, you need to specify the output functions of the inputs for all i

$$AO_i = F(ECI_1,..,ECI_n); \tag{4}$$

$$ECO_i = F(RI_1,..,RI_m), \tag{5}$$

From expressions (4) and (5) it is seen that the switching state is a special case of combinational state, when the function in the right part of these equations takes a constant value equal to the value of a certain input event.

In the automatic state, the functions μ_s, t_s describe two automata – activities and transactions, each of which is a tuple similar to the tuple (1)

$$\mu_s = <\boldsymbol{ECI}, \boldsymbol{AO}, \boldsymbol{S_\mu}, s_{\mu 0}, \mu_\mu, t_\mu>; \tag{6}$$

$$t_s = <\boldsymbol{RI}, \boldsymbol{ECO}, \boldsymbol{S_t}, s_{t0}, \mu_t, t_t>, \tag{7}$$

where ***ECI***, ***AO***, ***RI***, ***ECO*** are the sets described in expression (3); $\boldsymbol{S_\mu}$, $\boldsymbol{S_t}$ is the set of states of automatic machines of activity and transactions, respectively; $s_{\mu 0}$, s_{t0} is an initial state of automatic machines state of activity and transactions, respectively; μ_μ, μ_t is a function of outputs of automatic machines state of activity and transactions, respectively; t_μ, t_t is is a function of transactions of automatic machines state of activity and transactions, respectively. In this way we can describe the hierarchy in the semantic and logical model of automata/

We write the functions μ_s, t_s in the form of information arrays of size $m \times n$ (*MS*) and $l \times k$ (*TS*), where m, n, l, k are the power of the sets ***ECI***, ***AO***, ***RI, ECO***, respectively. Herewith

$$MS[i,j] = \begin{cases} 1, if\ ECI_i \Leftarrow AO_j \\ \quad 0, else \end{cases}, \tag{8}$$

where i, j are the numbers of the elements in the array, $i \leq m, j \leq n$.

$$TS[i,j] = \begin{cases} 1, if\ RI_i \Leftarrow ECO_j \\ \quad 0, else \end{cases}, \tag{9}$$

where i, j is the row and column number of the element in the array, $i \leq l, j \leq k$.

An example of a graphical interpretation of the causal relationship of the simple state of the control machine is shown in Figure 4.

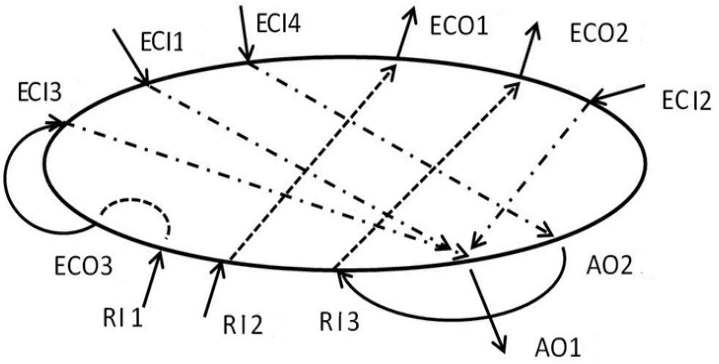


Figure 4: Graphical interpretation of the causal relationship of the simple state of the control machine

The set of ***ECI*** in Figure 4 is represented by inputs *ECI1*, *ECI2*, *ECI3*, *ECI4* (***ECI*** = {*ECI1*, *ECI2*, *ECI3*, *ECI4*}). Similarly, we describe the sets ***RI*** = {*RI1*, *RI2*, *RI3*}, ***ECO*** = {*ECO1*, *ECO2*, *ECO3*} and ***AO*** = {*AO1*, *AO2*}. The function of the state activity includes the following causal relationships: μ_s = {*ECI1* <= *AO1*, *ECI2* <= *AO1*, *ECI3* <= *AO1*, *ECI4* <= *AO2*} (event-driven activity). According to (8), this function is also described by an MS data array of size 3 × 3

$$MS = \begin{vmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}.$$

And the function of state transactions is represented by the following relations t_s = {*RI1* <= *ECO3*, *RI1* <= *ECO2*, *RI3* <= *ECO2*} (event controlled by the sensor) or in the form of a data array *TS* of size 4 × 2 which according to (9) has the form

$$TS = \begin{vmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{vmatrix}.$$

In Figure 4, the relationship of activities and transactions is represented by dashed and dotted lines, respectively. Note also that in the presence of an external connection *ECO3 – ECI3*, you can implement the ratio *RI1* <= *ECO3* <= *ECI3* <= *AO1* (the sensor controls the activity), using external connection *AO2 – RI3* ratio *ECI4* <= *AO2* <= *RI3* <= *ECO2* can be implemented (event driven event). Figure 4 shows a general case of the state structure. In some cases, the individual elements of the tuple (3) may be empty. For example, in the initial state there may be no *ECI*, and in the final – *ECO*. The empty set *AO* states are possible as well.

An example of a graphical interpretation of the causal relationship of the automatic state of the control machine is shown in Figure 5.

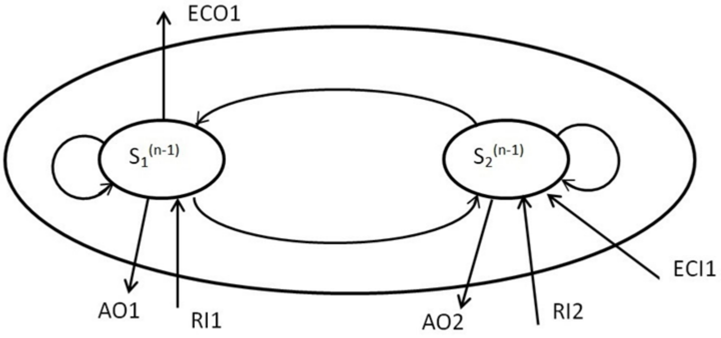


Figure 5: Graphical interpretation of the causal relationship of state of the automata

This state has a set of external connections *ECO1*, *ECI1*, *AO1*, *AO2*, *RI1*, *RI2*. Its internal structure is an automaton of the (n-1)-th level with two states $S_1^{(n-1)}$ and $S_2^{(n-1)}$. In the general case, the automatic state may represent more than one parallel operating control machines.

The process of building a model of causal relations in the system will be considered on the example of the temperature control system of the object [5, 6]. The control automata of the control unit in such a system performs the functions of a relay controller with thresholds of temperatures θ_{on} and θ_{off} ($\theta_{on} < \theta_{off}$). The control unit has two states: "HEAT" S_1 and "COOLING" S_2. State S_2 is initial ($s_0 = S_2$). The operating machines of the system generate the following events: $E1$ – if the current temperature $\theta > \theta_{on}$; $\neg E1$ – if $\theta \leq \theta_{on}$; $E2$ – if $\theta > \theta_{off}$; $\neg$E2 – if $\theta \leq \theta_{off}$.

Figure 6 shows a modified graph of the machine with additional arcs coming from the states. These are the arcs "state - action name - output operating machine - input of the control object" and "output of the control object input operating machine - reaction name - state input". Arcs, transitions from one state to another, are not directly related to the activities and values of the outputs of the control object.

The main functions of the machine shown in Figure 6 are as follows:

- The output function is represented with signals on, off;
- The transition function is represented with events E1, E2, ¬E1, ¬E2;
- The function of influences is represented by a controlled heat flow;
- The reaction function is represented by the temperature at the input of the TS sensor.

The states S_1 and S_2 of the automaton in Figure 6 are described by the tuple (3)

$$S_1 = <\mathbf{1ECI}, \mathbf{1RI}, \mathbf{1ECO}, \mathbf{1AO}, 1\mu_s, 1t_s>, \tag{10}$$

$$S_2 = <\mathbf{2ECI}, \mathbf{2RI}, \mathbf{2ECO}, \mathbf{2AO}, 2\mu_s, 2t_s>, \tag{11}$$

where ***1ECI***, ***1RI***, ***1ECO***, ***1AO*** are a set of interface elements of state S_1; ***2ECI***, ***2RI***, ***2ECO***, ***2AO*** are a set of interface elements of state S_2; $1\mu_s$, $1t_s$ is a function of activity and state transactions of state S_1, respectively; $2\mu_s$, $2t_s$ is a function of activity and state transactions of state S_2, respectively.

These sets consist of the following elements: ***1ECI*** = *<1ECI1, 1ECI2>*, ***1RI*** = *<1RI1, 1RI2>*, ***1ECO*** = *<1ECO1, 1ECO2>*, ***1AO*** = *<1AO1>*, ***2ECI*** = *<2ECI0, 2ECI1, 2ECI2>*, ***2RI*** = *<2RI1, 2RI2>*, ***2ECO*** = *<2ECO1, 2ECO2>*, ***2AO*** = *<2AO1>*.

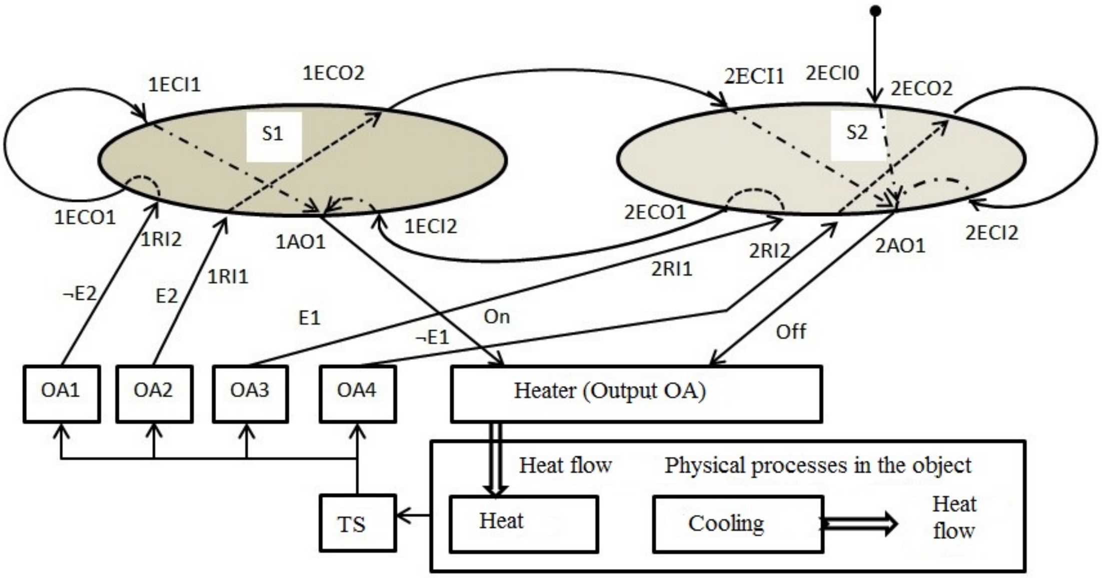


Figure 6: Graph of the control machine with contours of control and activity

In terms of activity, states S_1 and S_2 are combinational. We detail expression (4) using the logical OR function

$$1AO1=1ECI \vee 1EC2; \tag{12}$$

$$2AO1=2ECI0 \vee 2ECI1 \vee 1EC2. \tag{13}$$

In terms of transactions, states S1 and S2 are switching. We detail expression (5) using the following connections

$$1RI1 <= 1ECO2;\ 1RI2 <= 1ECO1; \tag{14}$$

$$2RI1 <= 1ECO1;\ 2RI2 <= 1ECO2. \tag{15}$$

The proposed causal relations of the states of the automaton allow to determine the causal relations in the contours of the automaton. For example, the following relationship results from diagram Figure 6:

1. The output *1AO1* is a consequence of any entry into state S_1.
2. If the active output *1AO1*, the input of the heater receives the command "on".
3. If the heater input receives the command "on", the heater generates heat flow to the control object.
4. If there is heat flow to the control object, its temperature rises.
5. If the temperature of the control object rises, sooner or later there will come a time when the output of the operating machine *OA2* will form an event *E*2 which is transmitted to the input *1RI1*.
6. The consequence of the event at the input 1RI1 is the event of transition to the state S_2, which is formed at the output *1ECO2*.
7. At any input to state S_2 according to (13) the command "Off" is formed at the input of the heater.

In cases where you want to focus on the management structure, it is advisable to show the outline of the generalization (see Figure 7).

The heating continuation circuit in this case contains: *1AO1* – Heating circuit – *1RI2* – *1ECO1* – *1ECI1* – *1AO1*, and the cooling switch control circuit: *1AO1* – Heating circuit – *1RI1* – *1ECO2* – *2ECI1* – *2AO1*.

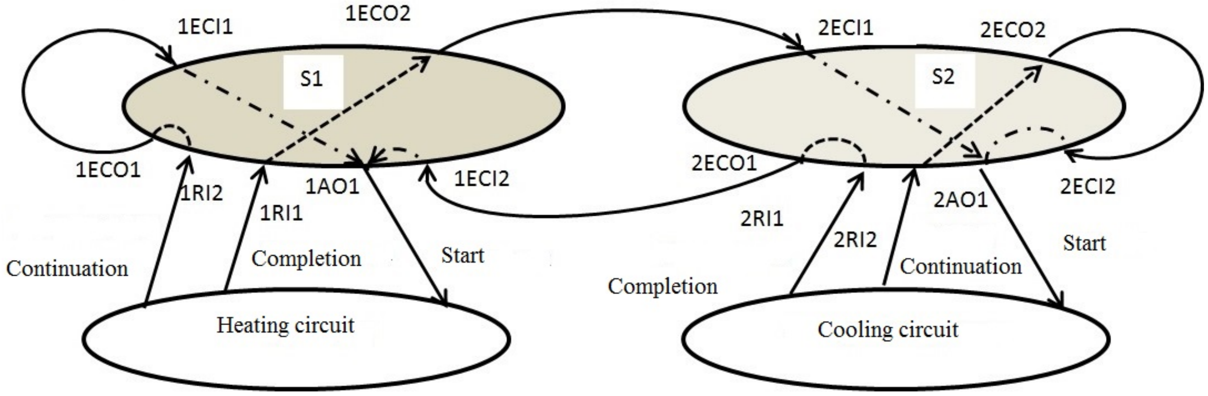


Figure 7: Graph of the control machine with generalized contours of activity

The above causal relationships in the contours of activity and control describe the behavior of control of the object of the system. Modern systems, for example, cognitive systems [7], have other behaviors, such as target, scenario, emergency, diagnostic and other behaviors that are associated with the behavior of controlling the object of the system.

As an example, let us consider the use of cause-and-effect relationships for diagnosing the technical state of the elements of the control device and changing the parameters of the object of the temperature control system. An enlarged block diagram of this system is shown in Figure 8.

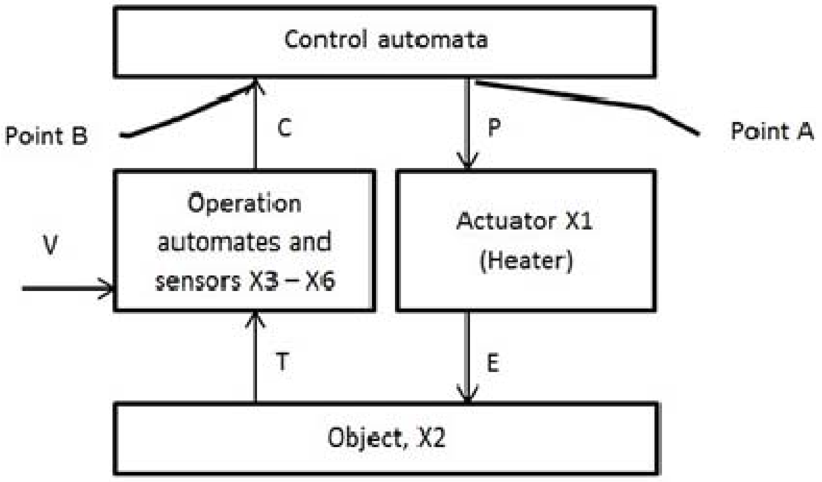


Figure 8: Enlarged block diagram of a temperature control system

Figure 8 uses the following notation:

P is a control command (on, off);

E is an energy (h (heat flux), nh (no energy));

T is a temperature change (inc (increasing), const (constant), dec (decreasing));

V is a time from begin of operation (little (), middle (), big ();

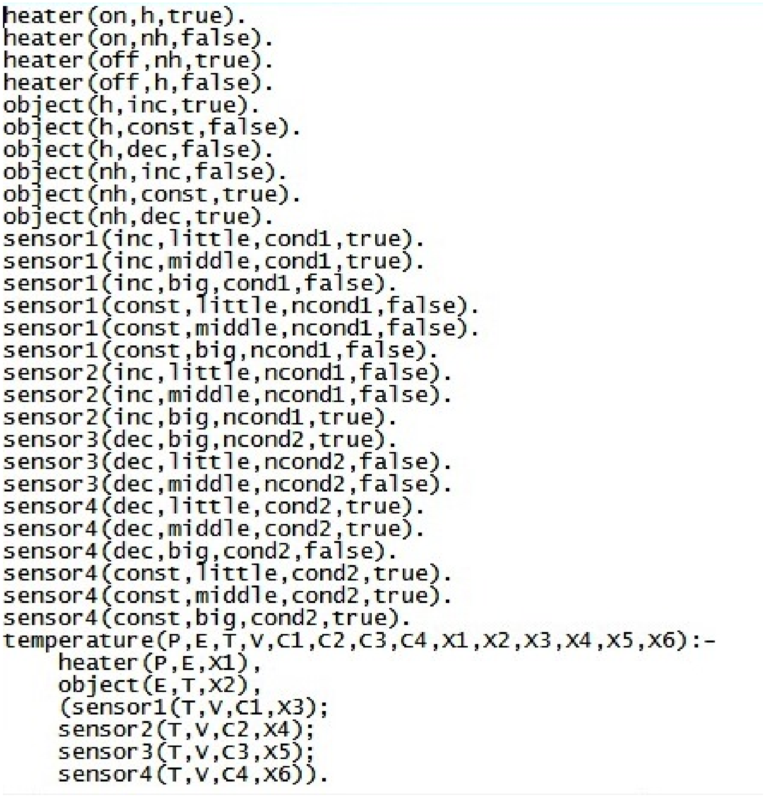
C is a heating / cooling conditions (C1 (cond1) – continue heating; C2 (ncond1) – finish heating; C3 (ncond2) – finish cooling; C4 (cond2) – continue cooling;

X is a serviceability (true) / malfunction (false) of the heater (X1), object (X2), operating machines (X3 – X6).

Based on the diagram in Figure 8, we describe the predicates of the input / output ratios with the assessment of the health of the node: heater (P, E, X1); object (E, T, X2); sensor1 (T, V, C1, X3), sensor2 (T, V, C2, X4), sensor3 (T, V, C3, X5), sensor4 (T, V, C4, X6). An example of the knowledge base of the system Figure 8 (facts and rules) is shown in Figure 9 as a fragment of the Prolog program of the SWI version.

The facts of this program describe the functioning of the system with the following simplifications:

- The heater has one level of output power;
- The rate of heating and cooling of the object is constant;
- The time markers of the experiment are chosen in such a way that the "big" time is sufficient to heat the object from θ_{on} to θ_{off} and its cooling from θ_{off} to θ_{on}, but the time "little" and "middle" is not enough;
- Events C1 – C4 are considered mutually exclusive.



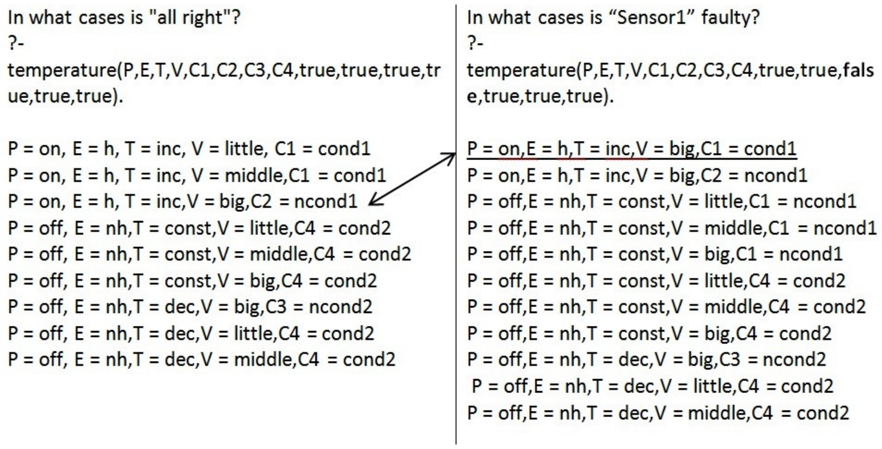
```
heater(on,h,true).
heater(on,nh,false).
heater(off,nh,true).
heater(off,h,false).
object(h,inc,true).
object(h,const,false).
object(h,dec,false).
object(nh,inc,false).
object(nh,const,true).
object(nh,dec,true).
sensor1(inc,little,cond1,true).
sensor1(inc,middle,cond1,true).
sensor1(inc,big,cond1,false).
sensor1(const,little,ncond1,false).
sensor1(const,middle,ncond1,false).
sensor1(const,big,ncond1,false).
sensor2(inc,little,ncond1,false).
sensor2(inc,middle,ncond1,false).
sensor2(inc,big,ncond1,true).
sensor3(dec,big,ncond2,true).
sensor3(dec,little,ncond2,false).
sensor3(dec,middle,ncond2,false).
sensor4(dec,little,cond2,true).
sensor4(dec,middle,cond2,true).
sensor4(dec,big,cond2,false).
sensor4(const,little,cond2,true).
sensor4(const,middle,cond2,true).
sensor4(const,big,cond2,true).
temperature(P,E,T,V,C1,C2,C3,C4,X1,X2,X3,X4,X5,X6):-
    heater(P,E,X1),
    object(E,T,X2),
    (sensor1(T,V,C1,X3);
    sensor2(T,V,C2,X4);
    sensor3(T,V,C3,X5);
    sensor4(T,V,C4,X6)).
```

Figure 9: An example of a system knowledge base in the form of a fragment Prolog program of the SWI version

In the experiments, the head of the rule with different variants of the predicate temperature is selected as the target of the Prolog program. The experimental results are shown in Figure 10.

As you can see from Figure 10, the results of the responses differ, which serves as a diagnostic sign of a malfunction. Opposite answers are marked with arrows, the answer typical for a faulty sensor 1 is underlined.

The program (figure 9) describes a chain of cause-and-effect relationships in the contour from point A to B (Figure 8), namely, B <= A. For example, from the command "to turn on the heater" (point A) it follows that the temperature of the object will rise, but for some time the event "continue heating" (point B) will remain true. On the other hand, from the event "continue heating", the command "to turn on the heater" follows. Accordingly, A <= B.

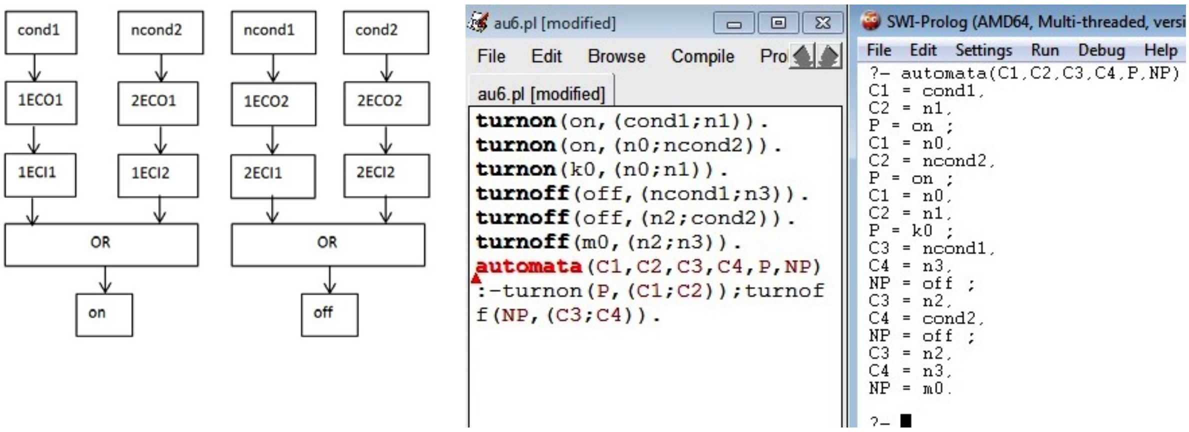


```
In what cases is "all right"?
?-
temperature(P,E,T,V,C1,C2,C3,C4,true,true,tr
ue,true,true).

P = on, E = h, T = inc, V = little, C1 = cond1
P = on, E = h, T = inc, V = middle,C1 = cond1
P = on, E = h, T = inc,V = big,C2 = ncond1
P = off, E = nh,T = const,V = little,C4 = cond2
P = off, E = nh,T = const,V = middle,C4 = cond2
P = off, E = nh,T = const,V = big,C4 = cond2
P = off, E = nh,T = dec,V = big,C3 = ncond2
P = off, E = nh,T = dec,V = little,C4 = cond2
P = off, E = nh,T = dec,V = middle,C4 = cond2
```

```
In what cases is "Sensor1" faulty?
?-
temperature(P,E,T,V,C1,C2,C3,C4,true,true,fals
e,true,true,true).

P = on,E = h,T = inc,V = big,C1 = cond1
P = on,E = h,T = inc,V = big,C2 = ncond1
P = off,E = nh,T = const,V = little,C1 = ncond1
P = off,E = nh,T = const,V = middle,C1 = ncond1
P = off,E = nh,T = const,V = big,C1 = ncond1
P = off,E = nh,T = const,V = little,C4 = cond2
P = off,E = nh,T = const,V = middle,C4 = cond2
P = off,E = nh,T = const,V = big,C4 = cond2
P = off,E = nh,T = dec,V = big,C3 = ncond2
 P = off,E = nh,T = dec,V = little,C4 = cond2
P = off,E = nh,T = dec,V = middle,C4 = cond2
```

Figure 10: Results of Experiments with the System Knowledge Base

The structure of the causal relationships of the automaton that controls the temperature of the object from its input events through its states and arcs to the outputs is shown in Figure 11a.



```
turnon(on, (cond1;n1)).
turnon(on, (n0;ncond2)).
turnon(k0, (n0;n1)).
turnoff(off, (ncond1;n3)).
turnoff(off, (n2;cond2)).
turnoff(m0, (n2;n3)).
automata(C1,C2,C3,C4,P,NP)
:-turnon(P,(C1;C2));turnof
f(NP,(C3;C4)).
```

```
?- automata(C1,C2,C3,C4,P,NP)
C1 = cond1,
C2 = n1,
P = on ;
C1 = n0,
C2 = ncond2,
P = on ;
C1 = n0,
C2 = n1,
P = k0 ;
C3 = ncond1,
C4 = n3,
NP = off ;
C3 = n2,
C4 = cond2,
NP = off ;
C3 = n2,
C4 = n3,
NP = m0.

?- █
```

a b c

Figure 11: The structure of the causal relationships (a) Prolog program (b) and modeling results (c) of the control automata

The knowledge base of this automaton in the form of a Prolog program in the SWI version is shown in Figure 11b. This program contains predicates **turnon**, **turnoff** and **automata** that associate input conditions (cond1, ncond1, cond2, ncond2) or their absence (n0, n1, n2, n3) with the automaton outputs (on, off) or their absence (k0, m0). Figure 11c shows the goal and simulation results, which confirm the dependence of the outputs of the machine on their inputs.

4. Conclusion

Traditional design technologies use formal models of control automata obtained as a result of the analysis of the verbal description of the control problem. The use of control automata narrows the knowledge base that is used by the system for control and reduces the quality of control under conditions of unpredictable changes in the system object, the environment and the technical state of the control device elements. The desire to take into account unused knowledge leads to the need to return to the verbal description of the problem, the identification and use of cause-and-effect relationships in the system, which are combined into the contours of activity and control through the object of the system.

The loops of activity in each active state support the conditions for the continuation of activity in this state, and the loops of control contain causal relationships for the completion of activities and transition to other states. The states of automata interact with contours through their interface elements: inputs and outputs of control events, outputs of activity and inputs of object reactions. A classification of states of an automaton is proposed, based on the logic of transformation of input elements of their interface into outputs, as well as a model of switching, combinational and automatic states.

The introduction of contours into control automata allows building chains of cause-and-effect relationships in the models of the system based on the principles of circular causality. The elements of these chains are described by the predicates "cause", "should" and others with arguments in the form of elements of the structure of the automaton or signals at its points. The arity of predicates can be expanded if we relate knowledge about the connections in the structure of the system with knowledge in the aspect of the research goal: usefulness, serviceability, testability, adaptability, reliability, and others. The knowledge base based on facts and rules with these predicates is explored by means of logical programming.

The proposed models have been tested on an object temperature control system. For this purpose, a knowledge base on the contours of systems has been developed in the form of facts and program rules in the Prolog language of the SWI version, describing the structure of cause-and-effect relationships in the system with the account the technical state of its elements. The program responses to the set goals contain diagnostic signs of malfunctions of system elements, such as the system object and its operating automata. In order to check the completeness of the behavior of the control automaton of the system, a knowledge base has been developed on the structure of causal-logical relationships in this automaton, which is also implemented as a program in the Prolog language of the SWI version. The simulation results confirmed the completeness of the behavior of the control automaton of the temperature control system.

The considered models of the contours of causality of links in the system are supposed to be used when assessing the usefulness of changes in the structures of hierarchical control automata in the processes of adaptation of the system.

5. References

[1] V. M. Glushkov, Synthesis of digital automata (Sintez tsifrovykh avtomatov), Fizmatizdat, Moscow, 1962.

[2] S. P. Eugene Xavier, Theory of automata, formal language and computation, New age international Pvt Ltd, New Delhi, 2008.

[3] V. N. Zakharov, D. A. Pospelov, V. Ye. Khazatskiy, Control systems. The task. Design. Implementation (Sistemy upravleniya. Zadaniye. Proyektirovaniye. Realizatsiya), Ed. 2nd, rev. and add., "Energy", Moscow, 1977.

[4] V.V. Rudnev, Finite automaton as a control object (Konechnyy avtomat kak ob"yekt upravleniya), Automation and Telemechanics, 9, 1978: 126–135.

[5] M. Poliakov, S. Subbotin, I. Andrias, Control System Control Unit FSM Semantic Models, System technologies. Regional interuniversity compendium of scientific works*,* Dnipro*,* 5 (2019): 43–53.

[6] M. Polyakov, A. Polyakov, State Models for FSM Semantic Models(Modeli sostoyaniy konechnogo avtomata dlya yego semanticheskoy modeli), in S.V. Morshchavka (Ed.), Abstracts reports of the X International scientific-practical conference of Modern problems and achievements in the field of Radio Engineering, Telecommunications and Information Technology, 07–09 October 2020, Zaporizhzhia [Electronic resource] Electron. data. – Zaporizhzhia, NU "Zaporizhzhia Polytechnic", 2020: 86–87.

[7] D.Vernon Artificial Cognitive Systems: A Primer, Cambridge, Massachusetts: The MIT Press, 2014.