

Common Sense Reasoning in Diagnostic Systems

Alexander P. Ereemeev and Vadim N. Vagin
Moscow Power Engineering Institute (Technical University)
Russia

1. Introduction

The problem of human reasoning modeling (so called “common sense” reasoning) in artificial intelligence systems and especially in *intelligent decision support systems (IDSS)* is very actual nowadays [Vagin et al., 2001]. That is why special attention is turned to case-based and analogous reasoning methods and models. The analogies and precedents (cases) can be used in various applications of artificial intelligence (AI) and for solving various problems, e.g., for diagnostics and forecasting or for machine learning. AI experts model case-based reasoning by computers in order to develop more flexible models of search for solutions and learning.

Investigation of mechanisms that are involved in the analogous reasoning process is an important problem for the specialists in AI. The analogy can be used in various applications of AI and for solving various problems, e.g., for generation of hypotheses about an unknown problem domain or for generalizing experience in the form of an abstract scheme. The great interest in this problem is caused by the necessity of modeling human reasoning (common sense reasoning) in AI systems and, in particular, in IDSS of real time.

Reasoning by analogy is to transfer of knowledge obtained from an object to a less studied one which is similar to the former with respect to some essential properties or attributes. Reasoning of this kind is a source of scientific hypotheses. Thus, analogy-based reasoning can be defined as a method that allows to understand a situation when compared with another one. In other words, an analogy is an inference method that allows to detect likeness between several given objects due to transfer of facts and knowledge valid for both objects, to other objects and to determine means of problem solution or to forecast unknown properties.

Case-based reasoning, like reasoning by analogy, is based on analogy; however, there are certain differences in their implementation. In the most encyclopedias, a precedent (from Latin, *precedentis*) is defined as a case that took place earlier and is an example or justification for subsequent events of this kind. Creating a precedent is to give grounds for similar cases in the future, and establishing a precedent is to find a similar case in the past.

The generalized structure of a real-time IDSS (RT IDSS) is represented in Fig. 1 [Vagin et al., 2007].

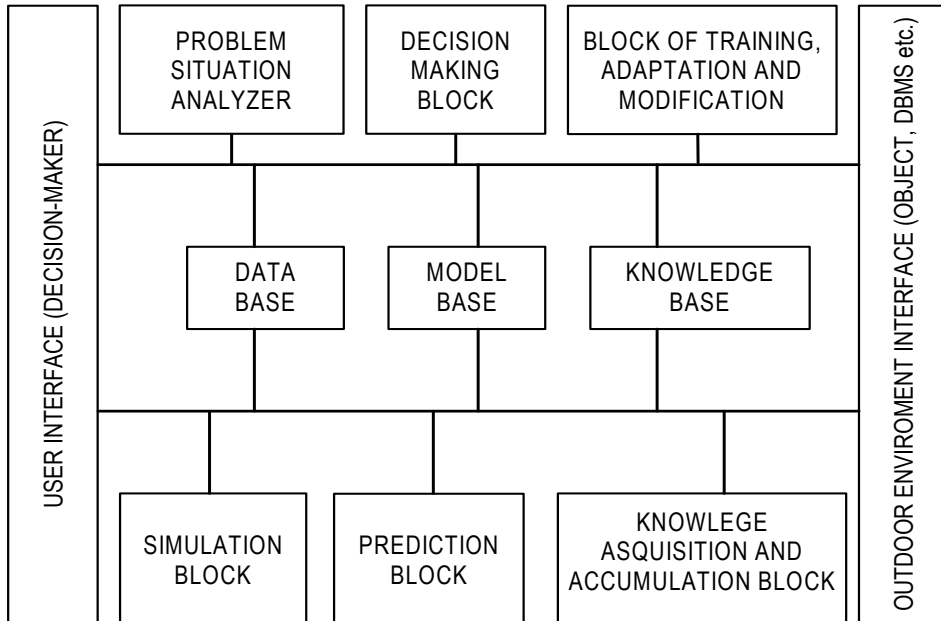


Fig. 1. The generalized structure of a real-time IDSS

Formally, a RT IDSS can be defined by the set

$$SS = \langle M, R(M), F(M), F(SS) \rangle,$$

where

$M = \{M_1, \dots, M_n\}$ is the set of formal or logic-linguistic models, implementing defined intelligent functions;

$R(M)$ is the function for selection of the necessary model in a current situation;

$F(M) = \{F(M_1), \dots, F(M_n)\}$ is the set of modification functions of models M_1, \dots, M_n ;

$F(SS)$ is the function for the SS modification, i.e. its basic components $M, R(M), F(M)$.

The main problems, solved by RT DSS, are:

- diagnostics and monitoring – revealing of problem situations;
- decision search – searching an optimal or admissible sequence of actions allowing to achieve the desired goal or solve the problem situations;
- forecasting – assessing the recommended actions related to the goal achievement (to solve a problem situation).

2. Reasoning by analogy

Nowdays there are a great number of various models, schemes, and methods that describe mechanisms of *reasoning by analogy* [Haraguchi et al., 1986; Long et al., 1994; Varshavskii et al., 2005; Ereemeev et al., 2005].

In (Haraguchi et al., 1986), the authors have proposed two types of analogies, *an analogy for solving problems* and *an analogy for forecasting*:

- *The analogy for solving problems* assumes the application of reasoning by analogy for increasing the efficiency of the problem solution which, generally speaking, can be solved without analogy as well as e.g., in programming and proving theorems;
- *The analogy for prediction (forecasting)* uses reasoning by analogy for obtaining new facts. Due to the transformation of knowledge based on the likeness of objects, one can make the conclusion that new facts probably hold.

Depending on the nature of information transferred from an object of analogy to the other one, *the analogy of properties and the analogy of relations can be distinguished*:

- *The analogy of properties* considers two single objects or a pair of sets (classes) of homogeneous objects, and the transferred attributes are the properties of these objects, for example, analogy between illness symptoms of two persons or analogy in the structure of the surfaces of Earth and Mars, etc.;
- *The analogy of relations* considers pairs of objects where the objects can be absolutely different and the transferred attributes are properties of these relations. For example, using the analogy of relations, bionics studies processes in nature in order to use the obtained knowledge in a modern technology.

We consider the methods of solution search on the basis of structural analogy which allows to take into account a context and based on the theory of structural mapping. We use semantic networks as a model of knowledge representation.

2.1 Reasoning by structural analogy taking into account the context

Consider an **analogy** as a quadruple $A = \langle O, C, R, p \rangle$, where O and R are the source object and the receiver object and C is the intersection object, i.e., the object that structurally is intersected with the source object and receiver object, and has a larger cardinality of the set of properties in the comparison with these objects. In other words, the analogy between the source object and receiver object is considered in the context of the intersection C , and p is a property for the definition of an original context. The structure of this analogy is represented in Fig. 2.

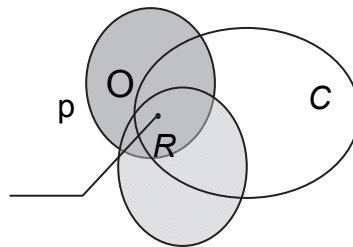


Fig. 2. Structure of analogy using the context

We use semantic networks (SNs) as a model of the knowledge representation for reasoning by analogy. The choice of an SN for the knowledge representation possesses an important advantage, which distinguishes it from other models, such as natural representation of structural information and fairly simple updating in a relatively homogenous environment. The latter property is very important for real-time IDSS oriented towards open and dynamical problem domains.

A *semantic network* is a graph $\langle V, E \rangle$ with labeled nodes and arcs, where V and E are sets of nodes and arcs, respectively. The nodes can represent objects (concepts, events, actions, etc.) of a problem domain, and the arcs represent relations between them.

By Pv , we denote the set of properties of an object $v \in V$.

Objects $v, v' \in V$ intersect each other on SN if and only if $Pvv' = Pv \cap Pv' \neq \emptyset$, where Pvv' is a set of common properties of objects v and v' .

By Vp , we denote a set of SN objects that have a property p .

By $Vv, Vv \subseteq V$, we denote an object set of objects that intersect $v \in V$.

The object C is an *intersection* for A if and only if there is $(C \in V) \ \& \ (p \in PC) \ \& \ (nR \leq nC) \ \& \ \neg(nR < nC) \ \& \ (nRC < nR) \ \& \ (nRC > 1)$, where nR and nC are the numbers of properties of the receiver R and the intersection C , respectively; nRC is the number of their common properties, $\neg(nR < nC)$ denotes that receiver R should not be much smaller than intersection C (i.e., the possibility of absorbing the receiver R by the intersection C , since here the probability of receiving a false analogy increases).

The object O is the *source* for analogy A if and only if there is $(O \in V) \ \& \ (p \in PO) \ \& \ (nO \leq nC) \ \& \ \neg(nO < nC) \ \& \ (nOC < nO) \ \& \ (nOC > 1)$, where nO is the number of properties of the source O ; nOC is the number of common properties of the source O and intersection C ; and other notations are analogous to the previous definition.

By $VC, VC \subseteq Vp$, we denote the set of objects that are candidates for the role of intersection C for analogy A .

By $VO \subseteq Vp$, we denote the set of objects that are candidates for the role of source O for analogy A .

By VA , we denote the set of analogies A .

The set $POCR = PO \cap PC \cap PR$ denotes *the context*, with respect to which analogy A is considered.

We consider the structure of the SN in detail (for Metalevel and for Situation 1) using the example from power engineering - operation control of the nuclear power unit (Fig. 3) [Eremeev et al., 2006a].

Let us give a semantic interpretation of the information given in the SN for Situation 1:

- It is recommended to supply the pump TH11D01 with boric concentrate 40g/kg caused by switching off automatic cooling system ACS 1 due to closing the gates TH11S24 and TH11S25;
- ACS 2 is switched off due to the closed gates TH12S24 and TH12S25;
- The upper setting T517B01 is equal to 63;
- The lower setting T517B01 is equal to 56;
- The upper setting TH11T500 is equal to 60;
- The lower setting TH11T500 is equal to 20.

Analogously, the fragments of the SNs, which illustrates Situations 2,3,4, are represented in Fig. 4.

2.2 Algorithm of reasoning by structural analogy

An SN with information about the problem domain, a receiver R , and the property for defining the original context p provide input data for this algorithm.

The algorithm for the problem solution on the basis of analogy taking into account the context consists of the following steps:

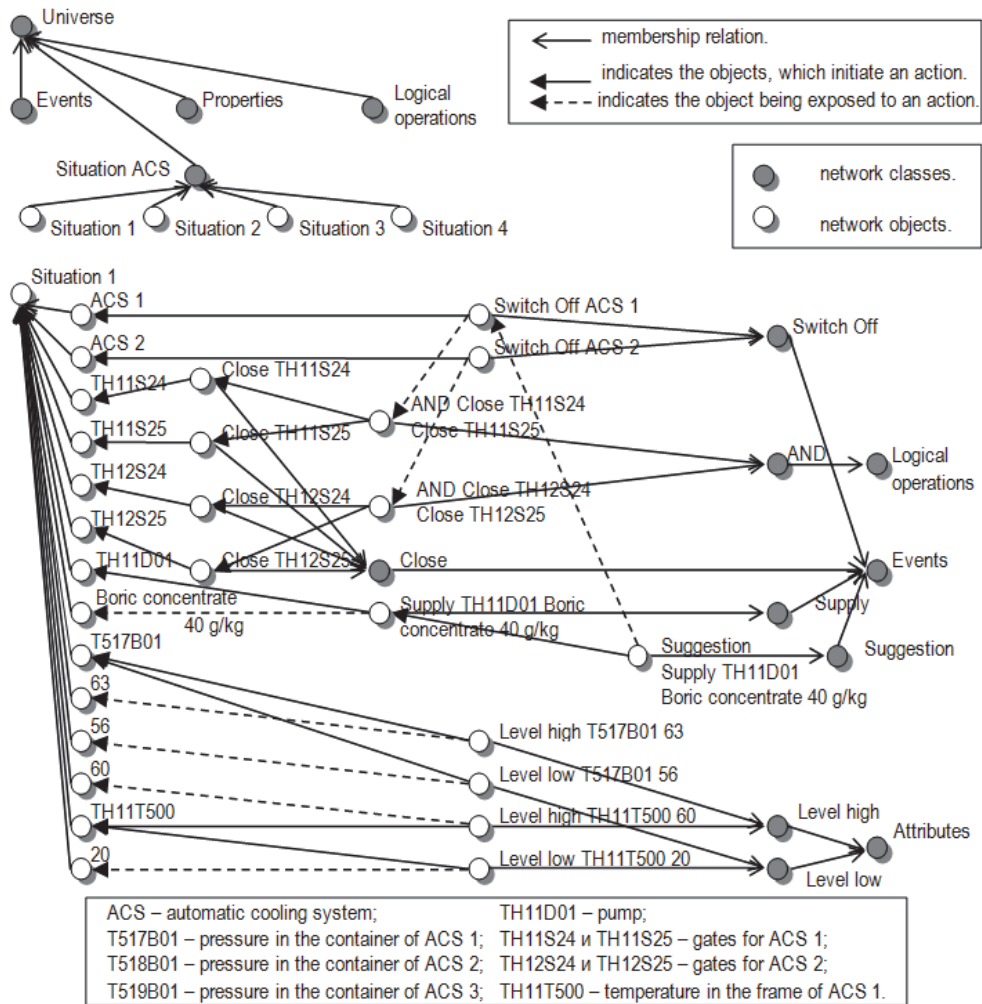


Fig. 3. A fragment of the SN that represents the Metalevel and the Situation 1 that was formed in the course of ACS functioning

Step 1. $VC = \emptyset$, $VO = \emptyset$, $VA = \emptyset$. Determine all objects of the SN, except for receiver R , that have property p ($Vp' = Vp \setminus \{R\}$). If there are no objects of this kind, then the search for a solution fails (without finding an analogy), otherwise, go to **step 2**.

Step 2. For the objects found in step 1, determine all possible intersections of C with R taking into account p (VC). If there are no intersections of C with R ($VC = \emptyset$), the first search for a solution fails, otherwise, go to step 3.

Step 3. From the objects extracted in step 1, determine all possible sources O for analogies (VO). In the case of success ($VO \neq \emptyset$), go to step 4, otherwise, the search for a solution fails.

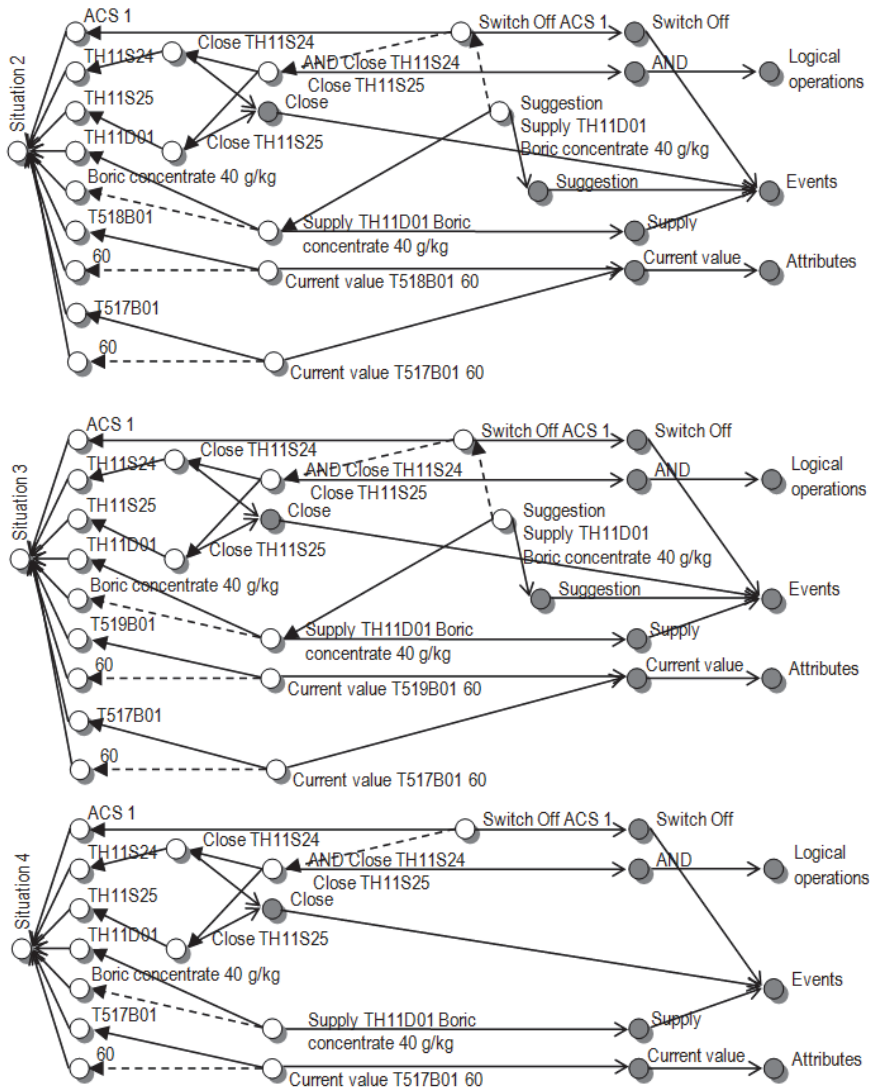


Fig. 4. The fragments of the SNs that represent Situations 2-4

Step 4. Construct possible analogies for R using the sets VC and VO . Add new analogy $A=\langle O,C,R,p \rangle$ to VA if and only if there exists an analogy $A'=\langle O',C,R,p \rangle$, $O \neq O'$. In the case of success ($VA \neq \emptyset$), go to step 5; otherwise, the search for a solution fails.

Step 5. The analogies obtained in step 4 (VA) (which could be previously compared with each other taking into account the context) are given to the decision making person (DMP), which means successful termination of the algorithm.

Having obtained analogies, the DMP may then make the final choice of the best ones. On the basis of these facts, the facts (properties) that hold for the source O are transferred to the receiver R .

Let us consider the steps of the functioning of the algorithm using the example from power engineering - operation control of the nuclear power unit (see Fig. 3,4).

As a receiver R for the analogy, we take Situation 4 and as the property p , we take Close TH11S24.

In the first step, $VC = \emptyset$, $VO = \emptyset$, $VA = \emptyset$ and $Vp' = \{\text{Situation 1, Situation 2, Situation 3}\}$. Since $Vp' \neq \emptyset$, we go to the next step.

Determine intersections of C with R taking into account p . Add in VC only Situation 1, because the number of common properties $n_{RC} = n_R$ for Situation 2 and Situation 3. Since $VC \neq \emptyset$, we go to the step 3.

Determine all possible sources O and go to step 4. In this case $VO = \{\text{Situation 2, Situation 3}\}$, because the Situation 1 is unique intersection for analogy.

In the fourth step, we construct only two analogies for R - Situation 4:

A1 = <Situation 2, Situation 1, Situation 4, Close TH11S24 >;

A2 = <Situation 3, Situation 1, Situation 4, Close TH11S24 >.

Add new analogies to VA and go to step 5.

The analogies obtained in step 4 ($A1$, $A2$) are given to the DMP.

As a result we obtain two analogies. Choosing one of them, the DMP can transfer facts that hold for the source of the analogy to its receiver. In this example, a new fact about the recommendation "Supply the pump TH11D01 with boric concentrate 40g/kg caused by switching off ACS 1 due to closing the gates TH11S24 and TH11S25" arises for Situation 4.

The methods of reasoning by analogy is more general than on the bases of cases. Analogies are used when it is impossible to find a suitable case in a case library. The reasoning by analogy method can be used independently from a case-based reasoning method as well as for correction (adaptation) of the nearest to a problem situation case to form a new case for completing a case library. Further we shall consider the case-based reasoning method and its application.

3. Case-based reasoning

Case-based reasoning (CBR), like analogous reasoning, is based on analogy; however, there are certain differences in their implementation [Aamodt, 1994; Ereemeev et al., 2006b].

As the practice shows, when a new problem situation arises, it is reasonable to use the method of case-based reasoning without drawing an analogy. This is caused by the fact that humans operate with these reasoning schemes at the first stages, when they encounter a new unknown problem.

Case-based reasoning is an approach that allows one to solve a new problem using or adapting a solution of a similar well-known problem. As a rule, case-based reasoning methods include four main stages that form a CBR-cycle, the structure of which is represented in Fig. 5.

The main stages of CBR-cycle as follows:

- Retrieving the closest (most similar) case (or cases) for the situation from the case library;
- Using the retrieved case (precedent) for solving the current problem;
- If necessary, reconsidering and adaptation of the obtained result in accordance with the current problem;

- Saving the newly made solution as part of a new case.

It is necessary to take into account that a solution on the basis of cases may not attain the goal for the current situation, e.g., in the absence of a similar (analogous) case in the case library. This problem can be solved if one presupposes in the CBR-cycle the possibility to update the case library in the reasoning process (inference). A more powerful (in detecting new facts or new information) method of reasoning by analogy is a method of updating case libraries. We also note that the elements of case-based reasoning may be used successfully in analogy-based reasoning methods; i.e., these methods successfully complement each other and their integration in IDSS is very promising.

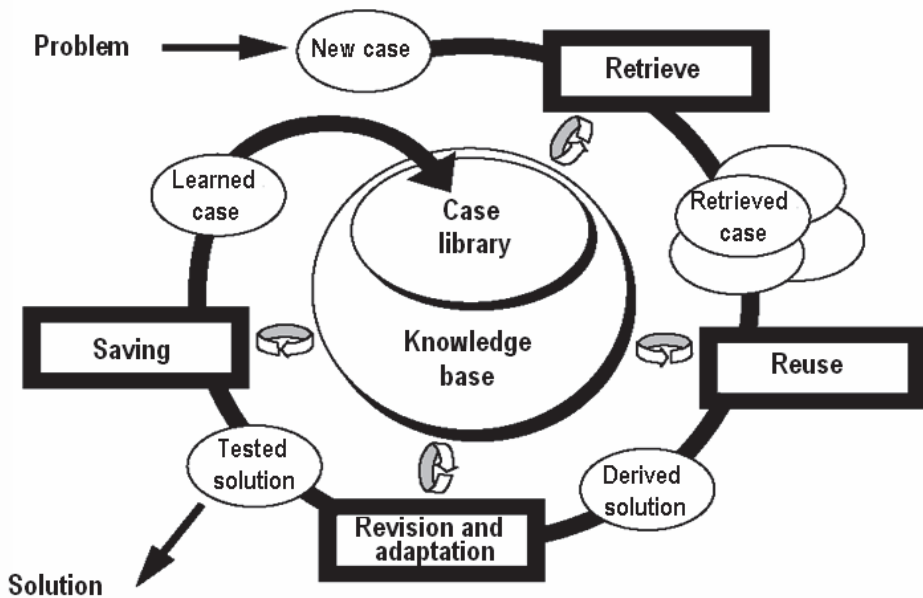


Fig. 5. The structure of CBR-cycle

Using the mechanism of cases for RT IDSS consists in issuing the decision to the operator (DMP – Decision Making Person) for the current situation on the basis of cases which are contained in a system. As a rule, the last stage in a CBR-cycle is excluded and performed by the expert (DMP) because the case library should contain only reliable information confirmed by the expert. Reconsidering and adaptation of the taken decision is required seldom because the same object (subsystem) is considered.

The modified CBR-cycle for RT IDSS includes following stages:

- Retrieving the closest (most similar) case (or cases) for the situation from the case library;
- Using the retrieved case (precedent) for solving the current problem.
- Case-based reasoning for IDSS consists in definition of similarity degree of the current situation with cases from a case library. For definition of similarity degree, the nearest neighbor algorithm (k-nearest neighbor algorithm) is used [Eremeev et al., 2007a].

Further, we shall view the structure of a case library for RT IDSS on the basis of non-classical logics for monitoring and control of complex objects like power units.

The case library for RT IDSS should join in itself the cases concerning a particular subsystem of a complex object, and also contain the information on each parameter which is used for the description of cases (parameter type and range). Besides, the case library should include such adjustments, as:

- the significance of parameter;
- a threshold value of similarity;
- a value which limits quantity of considered cases.

It is necessary to emphasize, that the case library can be formed on the basis of:

- the experience, accumulated by the expert;
- the analysis of the system archive;
- the analysis of emergencies;
- operative instructions;
- technological requirements.

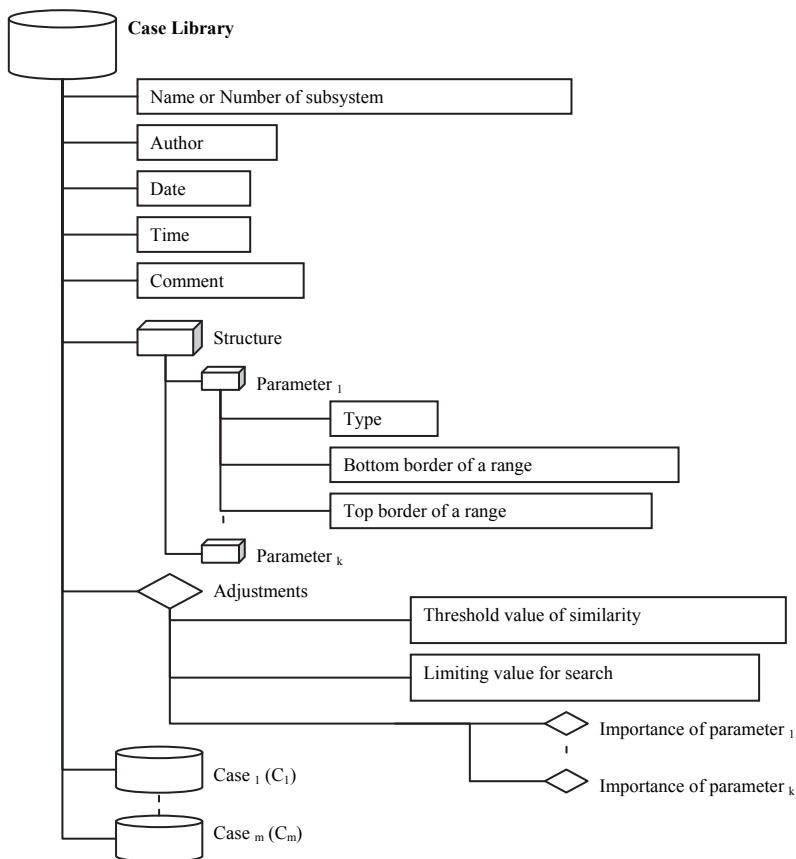


Fig. 6. The structure of the case library

The case library can be included in the structure of the knowledge base of RT IDSS or act as a separate component of the system. The structure of case library is represented in Fig. 6.

4. Application of case-based reasoning for diagnostics of complex object states

As a complex object, we shall view an object which has a complex architecture with various interrelations, a lot of controllable and operated parameters and small time for acceptance of operating actions. As a rule, such complex objects as the power unit are subdivided into technological subsystems and can function in various modes (regular, emergency, etc.).

For the description of such complex object and its subsystems, the set of parameters is used. The state of object is characterized by a set of concrete values of parameters.

In the operative mode, reading of parameters values from sensors for all objects is made by the system of controllers with an interval in 4 seconds. For this time interval, it is necessary to give out to the DMP (operator) the diagnosis and the recommendations on the given situation.

Diagnosing and detection of operating actions is carried out on the basis of expert knowledge, technological requirements and operative instructions. The developed software (Case Libraries Constructor – CLC) can be applied to the decision of the specified problems.

Basic components of CLC are:

- a module for storage and loading the cases to library and for data import;
- a subsystem of visualization for browsing the structure of a case library;
- a subsystem of editing and adjustment of a case library;
- a module of new case check;
- a subsystem of case library testing and case-based reasoning.

CLC was implemented in Borland C++ Builder 6.0 for Windows 7/ NT/XP.

This tool was applied in the prototype of a RT IDSS for monitoring and control of complex objects like power units on the example of a pressurizer in pressurized water reactor (PWR) of the atomic power station (Fig. 7) [Eremeev et al., 2007b, 2008].

Implementation of the case library with using CLC for expert diagnostics systems is subdivided into the following main stages:

- Creation of case libraries for subsystems of complex object;
- Adjustment of the created case libraries;
- Addition of cases in case libraries;
- Check of the added cases;
- Testing of the filled case libraries with using case-based reasoning;
- Reservation of the created case libraries for their subsequent transfer to operative maintenance.

5. Model-based diagnostics

Many diagnostics problems require building the behaviour prognoses, the work with contradictions and defaults, effective treatment of new facts and assumptions. The typical problem of diagnostics is to find a faults of a diagnosed device on the basis of some set of observations.

The generalized problem of diagnostic can be formulated as follows. There is a device exhibiting an incorrect behaviour. The device consists of components, one or several of

which are not working properly what is the reason of incorrect behaviour. There is a structure of connections between components and a possibility to get measurements on their inputs and outputs. It is necessary to determine what of components are faulty with minimal resource expenses. At present two main approaches to a solution of the given problem are viewed [Clansey, 1985; de Kleer et al., 1987; Forbus et al., 1993].

The first approach is heuristic diagnostics. The base of this approach is the knowledge extraction from an expert and building fault determining rules in the form of "symptoms \rightarrow faults" [Clansey, 1985].

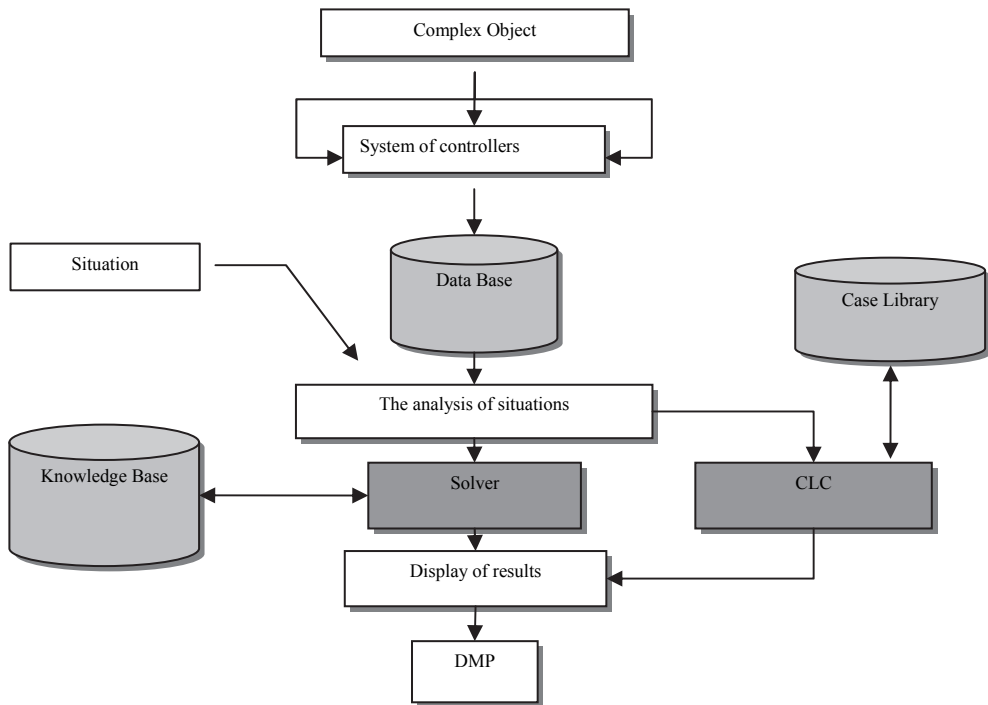


Fig. 7. The scheme of functioning for RT IDSS with use of CLC

Drawbacks of the given approach are:

- a complexity of multiple fault diagnostics;
- the problems in unexpected faults recognition;
- a complexity of expert knowledge formalizing;
- difficulties using the knowledge bases for other diagnostic problems;
- a rigid dependence from a device structure;
- a difficulty of explanations.

The second approach is model-based diagnostics [de Kleer et al., 1987; Forbus, 1993]. This approach is based on the knowledge of device component functionality.

The model of a device is a description of its physical structure, plus the models for each of its components. A compound component is a generalized notion including simple components, processes and even logical inference stages.

Model-based diagnosis process is the comparison of predicted device behaviour with its observed behaviour (Fig. 8).

It is supposed, that the model is correct, and all differences between device behaviour and a device model indicate availability of broken components.

Main advantages of the model-based approach:

- diagnosing the multiple faults;
- unexpected fault recognition;
- a precision of a component model description does not depend on the expert experience;
- a possibility of new device diagnosing;
- multiple using the models;
 - detailed explanations.

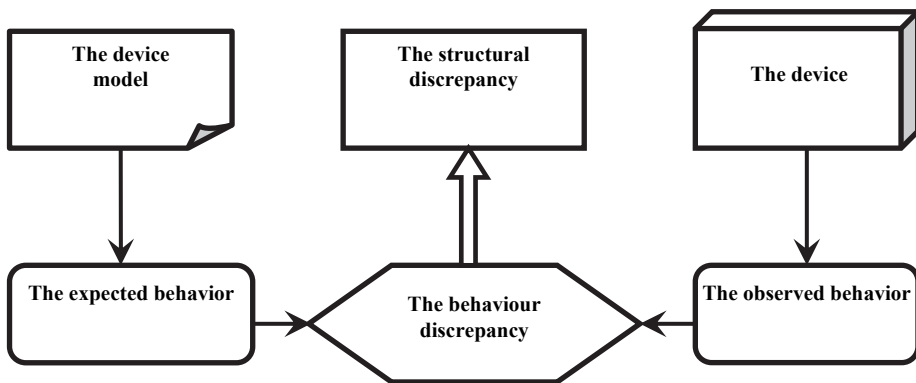


Fig. 8. Model-based diagnosis process

6. Assumption-based truth maintenance systems

For building a prognosis network, a component behaviour model, finding minimal conflicts characterizing irrelevance of observations with prognoses and minimal candidates for a faulty, it is profitable to use possibilities given by Assumption-based Truth Maintenance Systems (ATMS) [de Kleer, 1986; Vagin et al., 2008].

The Truth Maintenance Systems (TMS) are the systems dealing with the support of a coherence in databases. They save the assertions transmitted to them by a problem solver and are responsible for maintaining their consistency. Each assertion has the justification describing what kind of premises and assumptions this justification was obtained. The environment is a set of assumptions.

The inference of an inconsistency characterizes assumption incompatibility within the presuppositions of which this conclusion was made. Also there is introduced the environment set which contains some inconsistency [de Kleer, 1986]. The sets of inconsistency environments E_1, E_2, \dots, E_m are $\text{Nogood} = \{E_1, E_2, \dots, E_m\}$. A consistent ATMS environment is not Nogood.

There are the following correspondences between ATMS and the model-based diagnosis approach:

- ATMS premises - an observed device behaviour;
- ATMS assumptions - components of a device;
- inferred ATMS nodes - predictions of an diagnostic system;
- Nogood - the difference between predicted and observed device behaviour.

7. The current measurement point determination

One of the key aspects of the model-based fault search algorithm is to determine the optimal current measurement in a diagnosed device [de Kleer, 1987]. Efficiency of the current measurement choosing allows essentially reducing a decision search space while the inefficiency of choice will increase an operating time, the space of a searching algorithm, and also require additional resource spends to implement a measurement.

The best measurement point in a diagnosed device is a place (point) of measuring a value giving the largest information promoting the detection of a set of fault components at minimal resource spending.

One of the best procedures for reducing resource expenses is to produce the measuring giving the maximal information concerning predictions made on the basis of the current information on a system.

7.1 Heuristic methods of choosing a measurement point

The purpose of the best choosing a measurement point is to derive the maximal component state information. After each measuring there is a confirmation or refutation of prediction values in a point of measurement. So, it is possible to use the following aspects [Vagin et al., 2006a,b,c]:

- knowledge about environments that support predicted values in the measurement points which can be confirmed or refuted;
- knowledge about inconsistent environments;
- knowledge about coincided assumptions of the inconsistent environments.

7.2 Knowledge about supporting environments

The diagnostic procedure constructs predictions of values for each device point with the list of environments in which the given prediction is held. The list of environments represents assumption sets about correctness of corresponding device components.

Let's consider a sample of the 3-bit parity checker (Fig. 9) [Frohlich, 1998].

The vector (1, 0, 1) is input to device components (Inv1, Inv2, Inv3). Let the following measurements be obtained: D=0, R4=0, R2=1. The mismatch between observations and predictions speaks about a fault in a device. Based on measured observations additional predictions of values are formed. In general, it is obtained the following set of predictions with appropriate environments:

```
<R3=0, {{And3,Inv1}, {And2,And3,Inv6}, {And2,And3,Inv5}}>;
<R1=0, {{And1,Inv3}, {And1,And2,Inv4}, {And1,And2,Inv5}}>;
<O6=0, {{And2,Inv6}}>;
<O6=1, {{Inv3,Inv6}}>;
<O5=1, {{And2}}>;
<O5=0, {{Inv2,Inv5}, {And4,Inv1,Inv3,Inv4,Inv6}}>;
<O4=0, {{And2,Inv4}}>;
```

<O4=1, {{Inv1,Inv4}}>;
 <O3=1, {{And2}}>;
 <O3=0, {{Inv3}}>;
 <O2=0, {{And2,Inv5}}>;
 <O2=1, {{Inv2}, {And4,Inv1,Inv3,Inv4,Inv5,Inv6}}>;
 <O1=1, {{And2}}>;
 <O1=0, {{Inv1}}>.

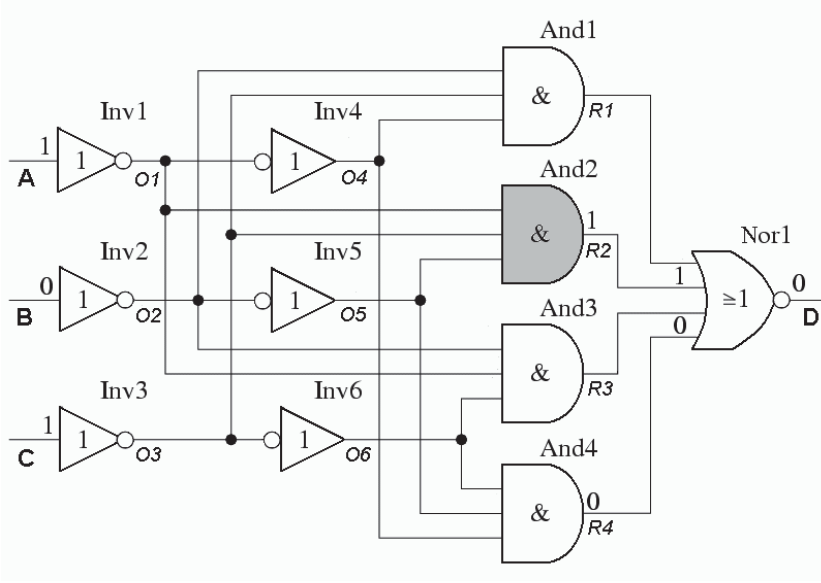


Fig. 9. The 3-bit parity checker

As we are interested with a measurement point with the greatest information on failure the point is selected from a quantity of assumptions.

Designate an environment set as $Env(x)$. So, for the considered example,

$Env(O1) = \{\{And2\}, \{Inv1\}\}$. Let's introduce the function $Quan(x)$, by which we will designate the information quantity obtained at measuring values in the point x .

If the environment J represents a unique assumption, then obviously the set cardinality will be equal 1: $|J| = 1$. The information quantity obtained from such environment is equal to 1. If the environment consists more than one component the information quantity obtained at confirming or refuting a value is less because we have knowledge not about a concrete valid / fault component but about a component set among of which are faulty. Therefore the information quantity obtained from an environment consisting of more than one assumption, we heuristically accept equal to half of set cardinality. Thus the function $Quan(x)$ is:

$$Quan(x) = \sum_{\substack{J_i \in Env(x) \\ |J_i|=1}} |J_i| + \sum_{\substack{J_j \in Env(x) \\ |J_j|>1}} \frac{|J_j|}{2} \quad (1)$$

Summing is produced on all possible values in the point x .

Then for the presented list of considered example predictions there are:

$$\text{Quan}(\text{R3}) = (2/2 + 3/2 + 3/2) = 4$$

$$\text{Quan}(\text{R1}) = (2/2 + 3/2 + 3/2) = 4$$

$$\text{Quan}(\text{O6}) = (2/2) + (2/2) = 2$$

$$\text{Quan}(\text{O5}) = (1) + (2/2 + 5/2) = 4.5$$

$$\text{Quan}(\text{O4}) = (2/2) + (2/2) = 2$$

$$\text{Quan}(\text{O3}) = (1) + (1) = 2$$

$$\text{Quan}(\text{O2}) = (2/2) + (1 + 6/2) = 5$$

$$\text{Quan}(\text{O1}) = (1) + (1) = 2$$

Points with the greatest value of the function $\text{Quan}(x)$ have the greatest priority of a choice.

We will call the given method of choosing a measurement point as SEH (Supporting Environment Heuristics).

In our example the point giving the greatest information quantity is O2.

7.3 Knowledge about the sets of inconsistent environment

As a result of each measurement there is a confirmation or refutation of some prediction. The environments E_1, E_2, \dots, E_m of refuted prediction form the set $\text{Nogood} = \{E_1, E_2, \dots, E_m\}$. It can be used for directional searching for more precise definition what kind of components from Nogood is broken.

Let we have the set $\text{Nogood} = \{\{\text{And2}, \text{Inv1}\}, \{\text{And2}, \text{Inv2}, \text{Inv5}\}, \{\text{And2}, \text{Inv3}\}\}$.

Obviously the more of the components from Nogood are specified by measuring a value in some device point, the more the information about which components of Nogood are broken will be obtained. For using this possibility, it is necessary to take the intersection of each environment from $\text{Env}(x)$ with each set from Nogood :

$$\text{Env}(x) \cap \text{Nogood} = \{A \cap B : A \in \text{Env}(x), B \in \text{Nogood}\}.$$

Continuing the example at Fig. 9, we obtain the following:

$$\langle \text{R3}, \{\{\text{Inv1}\}, \{\text{And2}\}, \{\text{And2}, \text{Inv5}\}\} \rangle$$

$$\langle \text{R1}, \{\{\text{Inv3}\}, \{\text{And2}\}, \{\text{And2}, \text{Inv5}\}\} \rangle$$

$$\langle \text{O6}, \{\{\text{And2}\}, \{\text{Inv3}\}\} \rangle$$

$$\langle \text{O5}, \{\{\text{And2}\}, \{\text{Inv2}, \text{Inv5}\}, \{\text{Inv1}\}, \{\text{Inv3}\}\} \rangle$$

$$\langle \text{O4}, \{\{\text{And2}\}, \{\text{Inv1}\}\} \rangle$$

$$\langle \text{O3}, \{\{\text{And2}\}, \{\text{Inv3}\}\} \rangle$$

$$\langle \text{O2}, \{\{\text{And2}\}, \{\text{And2}, \text{Inv5}\}, \{\text{Inv2}\}, \{\text{Inv1}\}, \{\text{Inv5}\}, \{\text{Inv3}\}\} \rangle$$

$$\langle \text{O1}, \{\{\text{And2}\}, \{\text{Inv1}\}\} \rangle$$

For this approach the equation (1) can be changed as follows:

$$\text{QuanN}(x) = \sum_{\substack{J_i \in \text{Env}(x) \cap \text{Nogood} \\ |J_i|=1}} |J_i| + \sum_{\substack{J_j \in \text{Env}(x) \cap \text{Nogood} \\ |J_j|>1}} \frac{|J_j|}{2}$$

For each of these points we calculate the information quantity as follows:

$$\text{QuanN(R3)} = 1 + 1 + 2/2 = 3$$

$$\text{QuanN(R1)} = 1 + 1 + 2/2 = 3$$

$$\text{QuanN(06)} = 1 + 1 = 2$$

$$\text{QuanN(05)} = 1 + 2/2 + 1 + 1 = 4$$

$$\text{QuanN(04)} = 1 + 1 = 2$$

$$\text{QuanN(03)} = 1 + 1 = 2$$

$$\text{QuanN(02)} = 1 + 2/2 + 1 + 1 + 1 + 1 = 6$$

$$\text{QuanN(01)} = 1 + 1 = 2$$

Points with the greatest value of function $\text{QuanN}(x)$ have the greatest priority of a choice. We will call the given method of choosing a measuring point as SIEH (Supporting and Inconsistent Environment Heuristics).

One can see that the point 02 as the most informative is again offered. And in the given approach the difference in information quantity between various points is expressed more brightly than without Nogood usage.

7.4 Knowledge about coincided assumptions of the inconsistent environments

During diagnostics of faulty devices as a result of confirmations and refutations of some predictions there is a modification of a set of inconsistent environments Nogood.

In each component set from Nogood one or more components are broken what was a reason of including a supporting set into the inconsistent environments Nogood. Taking the intersection of all sets of the inconsistent environments, we receive a set of components which enter into each of them, so their fault can be a reason explaining an inconsistency of each set holding in Nogood. Thus, we obtain the list of components a state of which is recommended to test first of all, i.e. the most probable candidates on faultiness.

The set intersection of inconsistent environments is expressed by the following equation:

$$\text{SingleNogood} = \bigcap_{E_i \in \text{Nogood}} E_i$$

For $\text{Nogood} = \{\{\text{And2, Inv1}\}, \{\text{And2, Inv2, Inv5}\}, \{\text{And2, Inv3}\}\}$ the set of the most probable candidates will be the following: $\text{SingleNogood} = \{\text{And2}\}$.

If $\text{SingleNogood} = \emptyset$, it means that there are some disconnected faults. In this case the given approach is inapplicable and it is necessary to define more precisely the further information by any other methods.

After obtaining a set $\text{SingleNogood} \neq \emptyset$, on the base of environments of value predictions in device points it is necessary to select those measurement points that allow to effectively test components to be faulted from SingleNogood .

For this purpose we will work with the sets obtained as a result of an intersection of each environment from $\text{Env}(x)$ with SingleNogood :

$$\text{Env}(x) \cap \text{SingleNogood} = \{J \cap \text{SingleNogood} : J \in \text{Env}(x)\}$$

The following versions are possible:

- a. $\exists J \in \text{Env}(x) : J \equiv \text{SingleNogood}$. One of environments of the value prediction in the point x coincides with the set SingleNogood . The given version allows to test faulty components from the set SingleNogood most effectively so this measurement point x is selected with the most priority.

- b. $\exists J \in \text{Env}(x): |J \cap \text{SingleNogood}| < |\text{SingleNogood}|$. The cardinality of *SingleNogood* is more than the cardinality of a set obtaining as result of an intersection *SingleNogood* with a set from *Env*(*x*). We evaluate this version as $\max_{J \in \text{Env}(x)} |J \cap \text{SingleNogood}|$ i.e. the more of components from *SingleNogood* are intersected with any environment from *Env*(*x*), the more priority of a choice of the given measurement point for the observation.
- c. $\exists J \in \text{Env}(x): \text{SingleNogood} \subset J$. The *SingleNogood* includes in a set from *Env*(*x*). We evaluate this version as $\min_{J \in \text{Env}(x)} (|J| - |\text{SingleNogood}|)$ i.e. the less a difference between *SingleNogood* and *Env*(*x*), the more priority of a choice of the given measurement point for the current observation.
- d. $\forall J \in \text{Env}(x): J \cap \text{SingleNogood} = \emptyset$, i.e. no-one of the most probable faulty candidates includes in environments *Env*(*x*) supporting predictions at the point *x*. We evaluate this version as the least priority choice, i.e. 0 in the numerical equivalent.

Also to the version (d) there are referred other methods of definition of current measurement point priorities which happen when *SingleNogood* = \emptyset . Thus in the estimations of a choice priority a numerical value returned as a result of call of other method is accepted. We call it by *ResultD*(*x*).

At appearance of the greater priority choosing between versions (b) and (c), heuristically we accept the version (b) as at this choice the refinement of faulty candidates is produced better. Note for various supporting sets of the same *Env*(*x*), the availability both the version (b) and the version (c) is also possible. In this case, as a resulting estimation for the given *Env*(*x*) the version (b) is also accepted.

Continuing the example, we obtain the following:

Nogood = {{And2,Inv1}, {And2,Inv2,Inv5}, {And2,Inv3}}

SingleNogood = {And2}

$$< R3, \text{version (c)}, \min_{J \in \text{Env}(x)} (|J| - |\text{SingleNogood}|) = 2 >$$

$$< R1, \text{version (c)}, \min_{J \in \text{Env}(x)} (|J| - |\text{SingleNogood}|) = 2 >$$

$$< O6, \text{version (c)}, \min_{J \in \text{Env}(x)} (|J| - |\text{SingleNogood}|) = 1 >$$

$$< O5, \text{version (a)} >$$

$$< O4, \text{version (c)}, \min_{J \in \text{Env}(x)} (|J| - |\text{SingleNogood}|) = 1 >$$

$$< O3, \text{version (a)} >$$

$$< O2, \text{version (c)}, \min_{J \in \text{Env}(x)} (|J| - |\text{SingleNogood}|) = 1 >$$

$$< O1, \text{version (a)} >.$$

Let's estimate the obtained results.

Designate by *maxd* the maximal numerical value among versions (d) for all assessed measurement points, and by *CompCount* a quantity of device components.

Accept in reviewing the following assessments:

1. $\max_{J \in Env(s(x))} |J \cap SingleNogood| < CompCount$. The quantity of components which are the intersection result is always less than the quantity of whole device components;
2. $\min_{J \in Env(s(x))} (|J| - |SingleNogood|) < CompCount$. The quantity of components in the prediction environment is always less than the quantity of the device components.

Taking into account these assessments, one can introduce a numerical assessment of the obtained results:

$$QuanSNG(x) = \begin{cases} 0, & \text{if } \forall J \in Env(s(x)) : J \cap SingleNogood = \emptyset \\ ResultD(x), & \text{if } SingleNogood = \emptyset \\ maxD + CompCount - \min_{J \in Env(s(x))} (|J| - |SingleNogood|), & \\ & \text{if } \exists J \in Env(s(x)) : SingleNogood \subset J \\ maxD + CompCount + \max |J \cap SingleNogood|, & \\ & \text{if } \exists J \in Env(s(x)) : |J \cap SingleNogood| < |SingleNogood| \\ maxD + 2 * CompCount, & \text{if } \exists J \in Env(s(x)) : J \equiv SingleNogood \end{cases}$$

Accordingly, for the example in the Fig. 9:

$maxD = 0$;

$CompCount = 11$;

$QuanSNG(R3) = 0 + 11 - 2 = 9$

$QuanSNG(R1) = 0 + 11 - 2 = 9$

$QuanSNG(O6) = 0 + 11 - 1 = 10$

$QuanSNG(O5) = 0 + 2*11 = 22$

$QuanSNG(O4) = 0 + 11 - 1 = 10$

$QuanSNG(O3) = 0 + 2*11 = 22$

$QuanSNG(O2) = 0 + 11 - 1 = 10$

$QuanSNG(O1) = 0 + 2*11 = 22$

The points with the greatest value of function $QuanSNG(x)$ have the greatest priority of choice. We will call the given method as SCAIEH (Supporting and Coinciding Assumptions of Inconsistent Environment Heuristics).

One can see that the most preferable measurement points are O1, O3 and O5, one of environment assumption of which coincides with *SingleNogood*. It differs from guidelines of other choice methods, but at the same time for the given example the choice of any of these points allows to test the most probable faulty component And2.

The developed methods of heuristic choice of the best current measurement point are recommended to use for devices with a great quantity of components as quality of guidelines directly depends on the quantitative difference of environments.

7.5 Practical results

Let's test the developed methods of the best measurement point choosing for the 9-bit parity checker [Frohlich, 1998].

For each experiment one of device components is supposed working incorrectly what is exhibited in a value on its output opposite predicted. A consequence of the incorrect component work is changing of outputs of those components which produce the results depending on values on the output of a faulty component. These changed results of component operations are transmitted to appropriate inquiries of a diagnostic system.

At the beginning of each experiment to inputs of components (Inv1, Inv2, Inv3, Inv7, Inv8, Inv9, Inv13, Inv14, Inv15) in a diagnostic complex the vector of values (1,0,1, 0,1,0, 1,0,1) enters. Then to the diagnostic system the value 0 retrieved from the output of the component Nor5 that depends on the work of a broken component and differs from predicted is transferred. It leads to the appearance of an inconsistency in the diagnostic system and starts the automatic process of testing.

In Fig. 10 the quantity of the stages required to each method for fault localization is shown. A method stage is a measurement point choosing. The smaller the quantity of method stages, the faster a fault is localized.

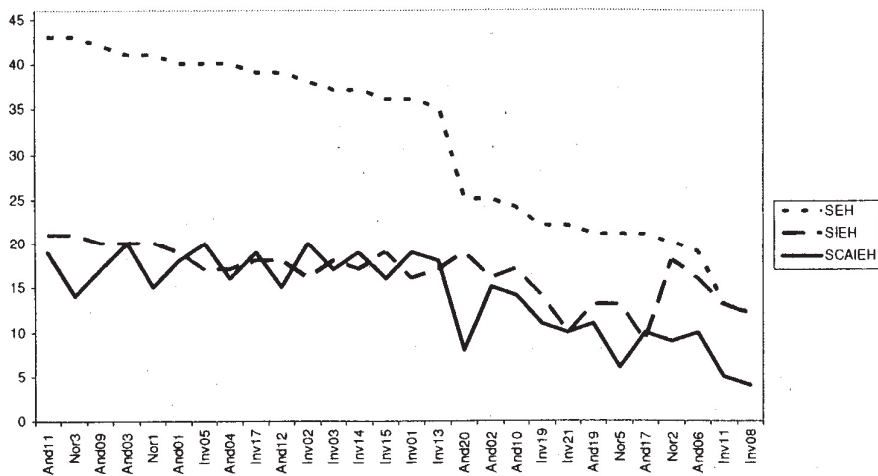


Fig. 10. The quantity of stages required to each method for fault localization

From the obtained results one can see that the method efficiency for different fault components is various and hardly depends on the device structure.

Let's estimate the method efficiency. The device consists of 46 components. The output values of 45 components are unknown (a value on the output of Nor5 is transmitted to the diagnostic system with input data together). So, the maximal stage quantity necessary for a fault definition is equal 45. Let's accept 45 stages as 100 %. For each experiment it is computed on how many percents each of the developed methods is more effective than exhaustive search of all values. Then define the average value of results. The evaluated results are represented in table 1.

The method	SEH	SEH	SCAIEH
On how many percents the method is more effective, %	30,79	63,17	68,65

Table 1. Results of experiments

From table 1 one can see that the greatest efficiency of current measurement point choosing has the heuristic method based on the knowledge about coincided assumptions of the inconsistent environments SCAIEH.

8. Conclusion

The method of reasoning by analogy on the basis of structural analogy was considered from the aspect of its application in modern IDSS, in particular, for a solution of problems of real-time diagnostics and forecasting. The example of the algorithm for solution search on the basis of analogy of properties that takes into account the context was proposed. This algorithm uses a modified structure of analogy that is capable of taking into account not one property (as in the base algorithm), but a set of properties. These properties determine the original context of analogy and transfer from the source to the receiver only those facts that are relevant in the context of the constructed analogy.

The method of case-based reasoning was considered from the aspect of its application in modern IDSS and RT IDSS, in particular, for a solution of problems of real-time diagnostics and forecasting. The CBR-cycle is considered and its modification for application in RT IDSS is offered. The k-nearest neighbor algorithm for definition of similarity degree of the current situation with cases from case library is described. The structure of case library for RT IDSS is proposed. The proposed method of case-based reasoning was implemented in Borland C++ Builder 6.0 for Windows 7/NT/XP. The main functional components of the implemented tool (Case Libraries Constructor – CLC) are specified.

The possibility of application of analogous reasoning in case-based reasoning is underlined. We also note that the elements of case-based reasoning may be used successfully in analogy-based reasoning methods; i.e., these methods successfully complement each other and their integration in IDSS is very promising.

The heuristic methods of finding the best current measurement point based on environments of device components work predictions are presented.

Practical experiments have confirmed the greatest efficiency of current measurement point choosing for the heuristic method based on the knowledge about coincided assumptions of the inconsistent environments SCAIEH.

Advantages of heuristic methods of the best current measurement point choosing is the simplicity of evaluations and lack of necessity to take into consideration the internal structure interconnections between components of the device.

The presented methods and tools were applied at implementation of a prototype of RT IDSS on the basis of non-classical logics for monitoring and control of complex objects like power units and electronic circuits.

9. References

- Vagin, V.N., Ereemeev, A.P., 2001. *Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making* // Journal of Computer and Systems Sciences International, v. 40(6), p.953-961.
- Vagin V.N., Yeremeyev A.P., 2007. *Modeling Human Reasoning in Intelligent Decision Support Systems* // Proc. of the Ninth International Conference on Enterprise Information Systems. Volume AIDSS. Funchal, Madeira, Portugal, June 12-16, INSTICC, pp.277-282.

- Haraguchi, M., Arikawa, S., 1986. *A Foundation of reasoning by Analogy. Analogical Union of Logic Programs* // Proceedings of Logic Programming Conference, Tokyo.
- Long, D., Garigliano, R., 1994. *Reasoning by analogy and causality: a model and application* // Ellis Horwood Series in Artificial Intelligence.
- Varshavskii, P.R., Ereemeev, A.P., 2005. *Analogy-Based Search for Solutions in Intelligent Systems of Decision Support* // Journal of Computer and Systems Sciences International, v. 44(1), p. 90-101.
- Ereemeev, A., Varshavsky, P., 2005. *Analogous Reasoning for Intelligent Decision Support Systems* // Proceedings of the XIth International Conference "Knowledge-Dialogue-Solution" – Varna, v.1, p. 272-279.
- Ereemeev, A.P., Varshavsky, P.R., 2006a. *Reasoning by structural analogy in intelligent decision support systems* // Proceedings of the VIIth Joint Conference on Knowledge-Based Software Engineering JCKBSE'06. IOS Press, pp. 303-306.
- Aamodt, E., 1994. *Plaza "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches"* // AI Communications, No. 7.
- Ereemeev, A., Varshavsky, P., 2006b. *Analogous Reasoning and Case-Based Reasoning for Intelligent Decision Support Systems* // International Journal INFORMATION Theories & Applications (ITHEA), Vol.13 № 4, pp. 316-324.
- Ereemeev, A., Varshavsky, P., 2007a. *Application of Case-based reasoning for Intelligent Decision Support Systems* // Proceedings of the XIII-th International Conference "Knowledge-Dialogue-Solution" – Varna, vol. 1, pp. 163-169.
- Ereemeev, A., Varshavsky, P., 2007b. *Methods and Tools for Reasoning by Analogy in Intelligent Decision Support Systems* // Proc. of the International Conference on Dependability of Computer Systems. Szklarska Poreba, Poland, 14-16 June, IEEE, pp.161-168.
- Ereemeev, A., Varshavsky, P., 2008. *Case-based reasoning method for real-time expert diagnostics systems* // International Journal "Information Theories & Applications", Volume 15, Number 2, pp. 119-125.
- Clancey W., 1985. *Heuristic Classification* // Artificial Intelligence, 25(3), pp. 289-350.
- de Kleer, J. and Williams, B.C., 1987. *Diagnosing multiple faults* // Artificial Intelligence, v.32, pp. 97-130.
- Forbus K.D., de Kleer, J. 1993. *Building Problem Solver* // A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.
- de Kleer J., 1986. *An Assumption-based TMS* // Artificial Intelligence, v.28, p.127-162.
- Vagin V.N., Golovina E.Ju., Zagoryanskaya A.A., Fomina M.V., 2008. *Exact and Plausible Inference in Intelligent Systems* / Vagin V.N., Pospelov D.A. (eds.) // M.: Fizmatlit, - 710 p. (in Russian).
- Vagin V.N., Oskin P.V., 2006a. *The Heuristic Methods of Obtaining the Effective Measurement in Diagnostic Systems* // Knowledge-based Software Engineering Proceedings of the 7th Joint Conference on Knowledge-based Software Engineering / E. Tyugu and T. Yamaguchi (eds). IOS Press, pp.275-284.
- Vagin V.N., Os'kin P.V., 2006b. *Heuristic and Probabilistic Methods for Taking Efficient Readings in Diagnostic Systems* // Journal of Computer and Systems Sciences International. V.1. 45, No 4, pp. 584 -598.
- Vagin V.N., Os'kin P.V., 2006c. *Multiagent Simulation Subsystem of Diagnostic Complexes Based on Device Models* // Journal of Computer and Systems Sciences International. Vol. 45, No 6, pp. 970-982.

Frohlich P., 1998. DRUM-II Efficient Model-based Diagnosis of Technical Systems // PhD thesis, University of Hannover.