Computer Assisted Mechanics and Engineering Sciences, **12**: 195–206, 2005.

Structural model and reasoning in hierarchical diagnosis

Jakub Oleksiak and Antoni Ligęza
AGH University of Science and Technology, Institute of Automatics,
Al. Mickiewicza 30, 30-059 Kraków, Poland

(Received March 31, 2005)

Fault diagnosis becomes more and more difficult and sophisticated task. This is so mainly due to growing complexity – contemporary technological systems are assembled from numerous components which cooperate and recursively include other components. The main goal of this paper consists in presentation of an approach which is able to reduce time of diagnosis and quantity of produced diagnoses by using hierarchical, logic-based approach. The reduction is achieved here due to two main factors. The first one is that a *hierarchical model* of systems is used. Such approach limits search space, because the system is considered at various levels of details and some diagnoses which are possible potential ones at more abstract levels can be verified to be impossible at more detailed levels. The second factor is that levels can be described with use of *different kinds of a logic-based knowledge representation*, what lets fit some best representation to a particular level.

1. Introduction

Diagnostic methodologies have emerged from numerous domains of Computer Science and Control Theory Korbicz et al. [5]. Some of these methodologies are based on models of diagnosed systems. In general, the models can be divided into the following groups:

- analytical models (Control Theory) e.g. physical equations, linear state equations Chow and Willsky, [2], state observers and Kalman filters Chen and Patton, [1], etc.,
- behavioral models (Computer Science) Patton and Korbicz, [14] e.g. expert systems Jagielski, [4]; Moczulski, [10], causal graphs Ligęza and Fuster–Parra, [8], fuzzy logic Kościelny et al. [6], etc.

In the paper we shall follow some of the basic notions introduced in another model-based diagnostic approach Reiter, [15]. Reiter defines a model as a set of components and relations among of them:

Definition 1. (Reiter). *A system is a pair (SD, COMPONENTS) where:*

- *SD, the system description, is a set of first-order sentences;*
- *COMPONENTS, the system components, is a finite set of constants.*

Note that all the presented before methodologies are usually flat; it means that after detection of a fault, diagnosis is carried out for the entire system. In the presented approach, a complex system is described by a hierarchical model where the position in vertical hierarchy represents some level of considered details. Hierarchical approaches for solving diagnostic problems were considered by a few authors e.g.: theoretical description of hierarchical approaches can be found in Giunchilia and Walsh, [3] and hierarchical diagnosis based on constraints were discussed in Mozetič, [11].

In this paper another, model-based hierarchical approach is considered. It is based on direct modeling of the hierarchy of system components which can recursively include other components.

The diagnostic procedure can refer to a certain level of component representation. In the classical Reiter's approach *COMPONENTS* are considered to be of atomic nature; here *complex* and *elementary components* will be introduced.

Generally, components can be described here by any kind of model-based diagnostic methodology which is able to produce diagnoses as sets of broken components; but in this article only two approaches are considered:

- A model-based one with Reiter's theory as a diagnostic procedure Reiter, [15],
- One using expert-based causal logical graphs with propagation of values in a expert graph as a diagnostic procedure Ligęza and Fuster–Parra, [8].

The model-based description does not require causal modeling, but it is computationally harder and not all technological components have an appropriate model useful in this kind of approach.

The expert graphs give possibility to create an efficient diagnostic description based on human causal knowledge about relations between causes of faults and their observed manifestations. The most important disadvantage is also lack of graphs for some components, and difficulties with building description for more complex systems with functional dependencies among variables. In these paper both of the approaches are combined to enable hierarchical diagnostic procedure.

Further, the causal AND/OR/NOT graphs for modeling relations among components will be used Ligęza and Fuster–Parra, [8], Ligęza, [9]. Causal graphs are defined as acyclic, directed graphs with nodes modeling logical functors. A type of a node of the graph can be: AND, OR or NOT. AND nodes represent logical conjunction of predecessor nodes, OR nodes represent disjunction of predecessor nodes and NOT arcs represent negation. AND nodes are represented in figures as nodes with an additional arc under the nodes, OR nodes are pure graph nodes, and NOT arcs are represented by arcs with black dot.

2. Hierarchical system model for automated diagnosis

The classical definition of *system* (1) introduced by Reiter, [15] puts forwards a *single-level* view on system modeling – the system is composed of a number of equally-ranked, atomic components, connected and interrelated. The behavior of the system is modeled with a *single-level*, flat first-order theory. No internal structure of components is considered.

For the sake of *hierarchical diagnosis* it is proposed to extend this definition over *complex systems*, recursively composed of components having some internal structure. A complex system CS is represented here by a set of top-level, interrelated components.

2.1. Elementary and complex components

Let there be given a set of constants *ELEMENTARY COMPONENTS* (EC, for short) denoting the basic components in the sense of Reiter, [15], i.e. ones denoted by constants for which no internal structure is ever considered. A (complex, hierarchically structured) component can be recursively defined as follows Oleksiak and Ligęza, [12]:

Definition 2. (Component). *A component c is either:*

- *an elementary component* $c \in EC$, *or*
- *a complex component* $c = (CD, SUBC)$, *where* CD *is the component description (a set of first–order sentences) and* $SUBC$ *is the set of its subcomponents.*

Note that the above definition introduces a *tree of component structure* modeling relations among components and their subcomponents (e.g. Fig. 1). Elementary components are leaves of the tree, they do not have any descendants. The root of the tree will be referred to as the *main component.* If $c = (CD, SUBC)$ is a component, then c is a *direct supercomponent* for any component $c_i \in SUBC$. Conversely, any $c_i \in SUBC$ is a *direct subcomponent* of c.

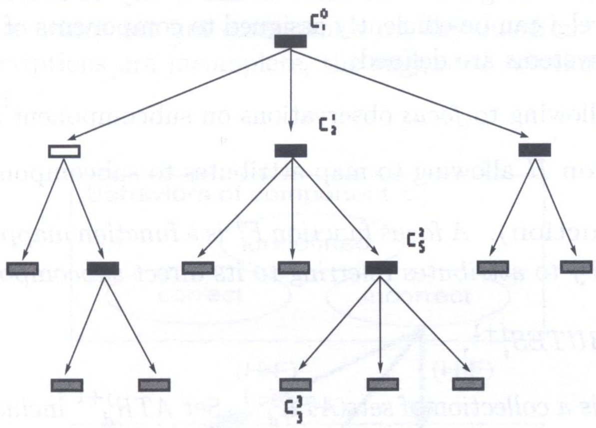

Fig. 1. The *tree of component structure* for a hypothetical system

We further assign the *level* to any node representing some components (subcomponents) in the tree. The root node (and the main component) is assigned a *level* numbered 0. Any other component is assigned level $m + 1$, where m is the level of its direct supercomponent. To denote the fact that component c_i is located at level j we shall write c_i^j.

Any component (either an elementary one or a complex one) can be further described by a set of variables, typically defining its inputs and outputs. The input and output variables, together with some other characteristics of the component (internal state variables, parameters, etc.) are also referred to as *component attributes.*

The hierarchical model, similarly to the flat model, has to involve some real world *observations.* The definition of observations is as follows:

Definition 3. *An observation of a complex component c_i^j is a finite set of first-order sentences. We shall write $(CD_i^j, SUBC_i^{j+1}, OBS_i^j)$ for a component with observation OBS_i^j.*

2.2. Diagnostic problem

Current observations can be *consistent* with predicted behavior of a component – and in such a case the component is believed to work correctly; this is denoted as $\neg AB(c_i)$ (recall that $AB(c_i)$ denotes the fact that component c_i behaves in an *ABnormal* way). If current observations are different from what can be expected on the basis of the component description, the component is assumed to be *faulty* – at least one subcomponent of it behaves in *abnormal* way; this is denoted as $AB(c_i)$. The diagnostic problem for the hierarchical model can be formulated as follows:

Definition 4. (Diagnostic Problem). *A diagnostic problem P_i^j for component c_i^j, defined at level j is defined as a four-tuple*

$$P_i^j = (c_i^j, CD_i^j, SUBC_i^{j+1}, OBS_i^j)$$

where CD_i^j is the model definition for component c_i^j, $SUBC_i^{j+1}$ are its subcomponents and OBS_i^j are the current observations at level j for component c_i^j.

Note that in the hierarchical approach, the diagnostic problem definition is always formulated at a certain *hierarchical level* and refers to some specific component.

2.3. Inter-level observations mapping

For enabling real hierarchical diagnosis one must define a way of moving a level down, so that information concerning level j can be efficiently assigned to components of level $j+1$. Two mapping functions for hierarchical systems are defined:

- **a focus function** F allowing to *focus* observations on subcomponent attributes,
- **a hierarchical function** H allowing to map attributes to subcomponent observations.

Definition 5. (Focus function). *A focus function* F_i^j *is a function mapping observations referring to a component* c_i^j *at level* j *to attributes referring to its direct subcomponents at level* $j+1$

$F_i^j : OBS_i^j \xrightarrow{P} ATTRIBUTES_i^{j+1}$,

where $ATTRIBUTES_i^{j+1}$ *is a collection of sets* ATR_d^{j+1}. *Set* ATR_d^{j+1} *includes attributes of subcomponents included in diagnosis d. Diagnosis d solves diagnostic problem* P_i^j.

Definition 6. (Hierarchical function). *A hierarchical function* H_k^{j+1} *is a function mapping attributes referring to a subcomponent* c_k^{j+1} *at level* $j+1$ *to its observations*

$H_k^{j+1} : ATR_k^{j+1} \rightarrow OBS_k^{j+1}$.

Hierarchical function is mainly used for conversion of the form of diagnostic information between hierarchical levels. Figure 2 presents both of the mapping functions in an intuitive, graphical form.

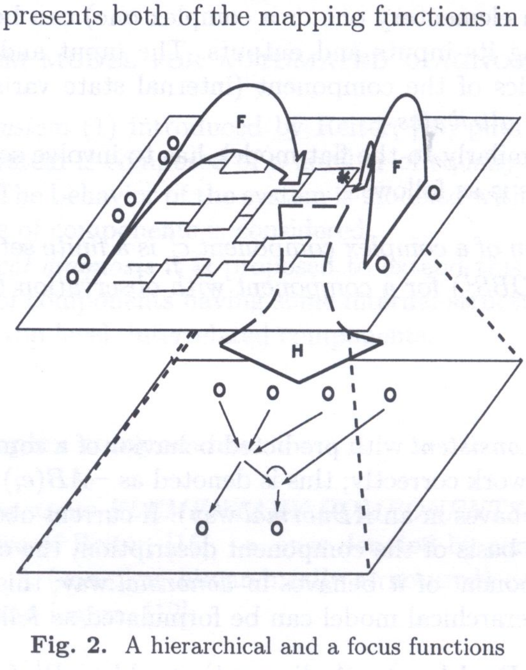

Fig. 2. A hierarchical and a focus functions

Let c_k^{j+1} be some direct subcomponent of c_i^j. The observations referring to c_k^{j+1} will be denoted as OBS_k^{j+1}, and there is $OBS_k^{j+1} \in H \circ F(OBS_i^j)$. If subcomponent c_k^{j+1} is diagnosed as faulty on the basis of observations OBS_i^j at level j, then observations OBS_k^{j+1} resulting from mapping the observations down to level $j+1$ can fall into one of the following three categories (see Fig. 3):

- $(H \circ F)''$ – observations which are consistent with description of correct behavior of subcomponent c_k^{j+1}; the description of incorrect behavior of component c_i^j is too general. This case is similar to the TC abstraction Giunchiglia and Walsh, [3]. The diagnosis is refused as false,
- $(H \circ F)$ – observations which are consistent with description of incorrect behavior of subcomponent c_k^{j+1}. In this case the diagnostic procedure can be continued on lower levels of hierarchy,
- $(H \circ F)'$ – observations which are not consistent with any one from the two mentioned before descriptions. The descriptions are incomplete, the suggested solution is to stop the diagnostic procedure at this level.

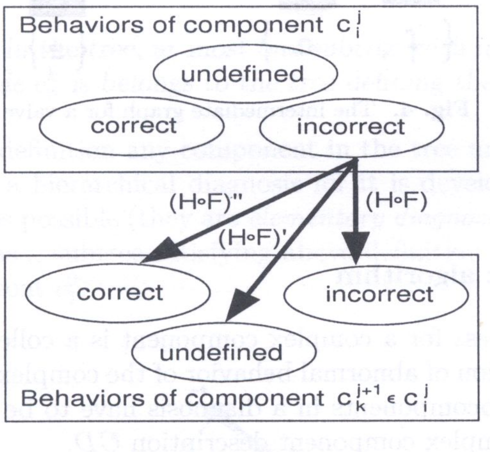

Fig. 3. Fault mapping

3. Hierarchical diagnostic algorithm

The presented algorithm is an implementation of the hierarchical diagnostic methodology and is composed of a *global diagnostic procedure* and a *local diagnostic procedure.*

3.1. The local diagnostic algorithm

The local diagnostic algorithm diagnoses components by using proper methodology for their diagnostic description CD. Two kinds of diagnostic descriptions can be analyzed by the local diagnostic algorithm: causal AND/OR/NOT graphs – diagnosing by propagation of states, and model-based approach – diagnosing by consistency-based procedures.

The result of the local diagnosis is an *intermediate graph.* The main reason for use of a graph form instead of a simple collection of diagnoses is that a graphical form is more readable for users. The result of the local diagnostic procedure is composed as disjunction of all diagnoses for the component where each diagnosis is represented by conjunction of faulty subcomponents.

The notation of graph nodes is similar to notation which is used for causal graphs. OR nodes are ordinary graph nodes; they represent logical disjunction of descendant nodes. AND nodes are nodes with an additional arc under the node; they represent logical conjunction of descendant nodes. Leaves of graphs are labeled by component names.

When it is necessary to calculate diagnoses from *an intermediate graph* then the state of its root node is set to "true". Further calculations are done according to state propagation rules in causal AND/OR/NOT graphs. If state of a leaf is "true", the component, whose name is on the label, is considered as faulty and becomes a part of a diagnosis.

The *intermediate graph* for a hypothetical valve is presented in Fig. 4. Diagnoses for the system are: $d_0 = \{f, a\}$ and $d_1 = \{e\}$.

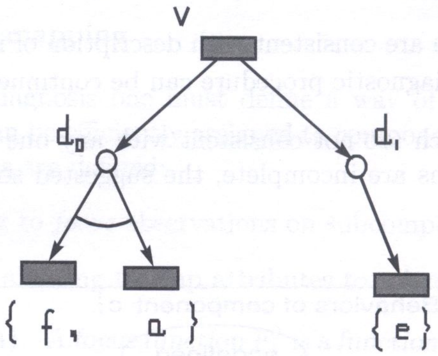

Fig. 4. The intermediate graph for a valve

3.2. The global diagnostic algorithm

The result of diagnostic process for a complex component is a collection of alternative diagnoses. Each diagnosis is an explanation of abnormal behavior of the complex component and includes some of its subcomponents. All subcomponents in a diagnosis have to be faulty for keeping consistency between observations and complex component description CD.

The logical aspect of hierarchical diagnostic reasoning requires that the following formula is recursively satisfied:

$$AB(c_i^j) = \bigvee_{d_k \in D_i} \left(\bigwedge_{c_n^{j+1} \in d_k} AB(c_n^{j+1}) \right).$$

Recursion of the formula is finished when either the collection of diagnoses for a complex component is empty or a component is an elementary component. More precisely, the diagnostic procedure is stopped and returned in the form of a *tree of diagnoses* when:

1. A diagnosed component is an elementary component. The algorithm turns back to its supercomponent and the component is recognized as an atomic element of a diagnosis.
2. It is impossible to establish values for suitable number of attributes or observations for a component. The component is treated as an elementary component.
3. Local diagnostic procedure is finished without any diagnosis. This happens if either observations generate a conflict in a causal graph or observations are not generating any conflict in a model. The algorithm turns back to supercomponent, the currently examined diagnosis is skipped, and the next diagnosis in order is analyzed.
4. The established values of some attributes are in conflict with either domains of attributes or other conditions imposed on attributes. The algorithm turns back to its supercomponent, the current diagnosis is skipped, and the next diagnosis in order is analyzed.

The final result of diagnostic process can be shown in visual form as a simple *tree of diagnoses* T_R (e.g. Fig. 5). Each complex component c_i^j labels a *component node*; the node is an *OR* node. Each diagnosis d_k of complex component c_i^j is represented by a descendant node of the *component node*. Nodes labeled by diagnoses are *diagnosis nodes* and they are *AND* nodes.

Each subcomponent of component c_i^j which belongs to a diagnosis is a descendant of the *diagnosis node*. *Diagnosis nodes* can be *expanded* to *component nodes* and *component nodes* can be expanded to further *diagnosis nodes* (if diagnoses exist). The highest complex component in hierarchy is represented by a root of T_R. The rest of the tree is built recursively down according to results of diagnosis of complex components. Consider a tree modeling the hierarchical diagnostic procedure, such as one presented in Fig. 5. Let us introduce the following definition:

Definition 7. *A hierarchical diagnosis for any component c_i^j at level j is any subtree satisfying the following conditions:*

- *the root of the subtree is any of the diagnosis nodes located directly under component node c_i^j,*

- *for any component node c_n^k in the tree, at most one subtree with its root node being a diagnosis node for the component node c_n^k is belongs to the tree defining the hierarchical diagnosis.*

With respect to the above definition any component in the tree modeling hierarchical diagnosis can be expanded (down) and a hierarchical diagnosis for it is developed in this way; for the leaf nodes, however, no expansion is possible (they are *elementary diagnoses*). The hierarchical diagnosis for the whole system is built as a subtree satisfying above definition and with root diagnosis node located directly under component c_1^0.

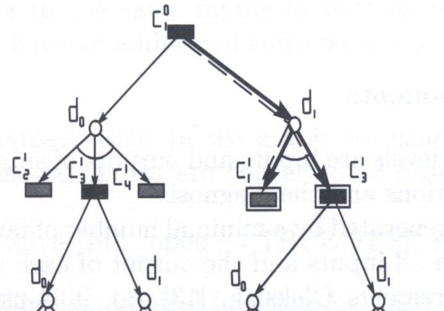

Fig. 5. Tree of diagnoses T_R with marked hierarchical diagnosis $\{c_1^1, c_3^1\}$

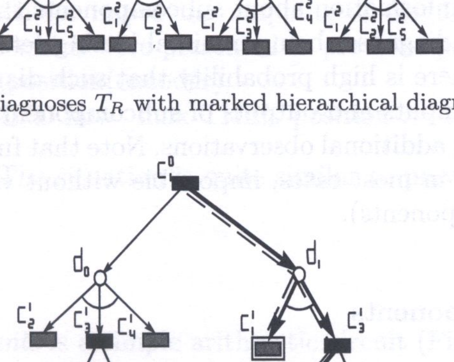

Fig. 6. Tree of diagnoses T_R with marked hierarchical diagnosis $\{c_1^1, c_1^2, c_3^2, c_4^2\}$

Note that for a given component, various diagnoses can be defined with respect to the degree of expansion. For example, there are two different hierarchical diagnoses marked in Figs. 5 and 6.

The global diagnostic procedure carries on diagnosis of the whole system and the diagnosis is based on results obtained from the local diagnostic procedure. The global diagnostic algorithm is based on top-down search in the *tree of diagnoses*. Such a search has two basic versions: depth-first search and breadth-first search.

The simplest sketch of the depth-first searching procedure is as follows:

1. `i = 1,    j = 0.`

2. $P_i^j \rightarrow D_i^j$.

3. SELECT p, k THAT $c_k^{j+1} \in d_p$ IS NOT PROCESSED AND $d_p \in D_i^j$.

4. IF k EXISTS THEN $j = j + 1$, $i = k$ AND GOTO 2.

5. IF $j = 0$ THEN EXIT.

6. IF k NOT EXISTS THEN $j = j - 1$ AND GOTO 3.

The diagnoses for the whole complex system can be shown when the diagnostic procedure is finished. The limitation of complexity by hierarchization is possible for all types of refutation procedures, if the number of outputs of components is limited and lower than the number of components at a diagnosed level Oleksiak, [13]. Otherwise, the situation depends on complexity of the original refutation procedure.

4. The focus function

4.1. Model-described components

Attributes for model-described levels are inputs and outputs of subcomponents. Values of attributes depend on: the model, observations and the diagnosis.

If we assume that faults are generated by a minimal number of faulty subcomponents from conflict sets (minimal hitting sets) then all inputs and the output of each subcomponent will be calculated without any additional measurements Oleksiak, [13]. So, it is useful for components where it is impossible to get any further information about subcomponents than values of inputs and outputs of the components. Moreover, diagnoses being minimal hitting sets are usually consistent with real reasons of malfunctions; so, there is high probability that such diagnoses are correct.

Otherwise, values of some inputs and outputs of subcomponents cannot be determined and the values have to be completed by additional observations. Note that further analysis of subcomponents on lower hierarchical levels is, in most cases, impossible without values of all attributes (values of inputs and outputs of subcomponents).

4.2. Graph-described components

Observations and intermediate symptoms taking part in diagnosis of a graph-described subcomponent s_p, become directly attributes of the subcomponent.

Let the list H_d include history of state propagation in a causal graph G for a diagnosis d. Then it is possible to determine all nodes which take part in creation of diagnosis d and they are either manifestation symptoms or intermediate symptoms. The nodes (symptoms) constitute a set P_d and they become attributes of each subcomponent included in diagnosis d.

Graph description has one unpleasant property: a lot of diagnostic information is lost during mapping through graph-described components. Deficit of information is especially harmful when subcomponents are described by their models. If values of some attributes are unknown or not precisely evaluated then not all the conflicts are generated for model-described subcomponents, since some ways of calculation of conflicts become unavailable. When not all conflict sets are generated then not all diagnoses are generated, moreover some diagnoses can be incorrect.

It is possible to carry out such diagnosis but it is also necessary to be aware that the set of diagnoses can be incomplete. One can propose some approaches to keep information flow among the distinguished levels of hierarchy at a satisfactory level of precision:

- additional measurements and interaction with users,
- guaranteeing that there are no model-described components below graph-described components in the hierarchical model,
- parameterization of subcomponents by typical values for certain kinds of faults,
- use of some additional knowledge holders for graph-described components Oleksiak, [13].

5. The hierarchical function

The hierarchical function maps attributes to observations. The role of this function is usually auxiliary and reduced to conversion of diagnostic information from one form to another. We can distinguish here four possible configurations of two hierarchical levels and two types of description:

- Model-model configuration. The hierarchical function maps attributes which represent a subcomponent inputs or outputs to the same inputs or outputs on the lower level of hierarchy. Sometimes, it is necessary to have an additional conversion, e.g. an integer value is divided into bits,
- Model-graph configuration. Observations in the graph are usually propositional symbols with boolean values, subcomponent attributes are converted to observations by pre-specified rules e.g.
  IF flow(valve1) > 15 THEN valve_full_open := TRUE ELSE valve_full_open := FALSE.
- Graph-model configuration. The hierarchical function for this case depends on additional solutions used with focus function for improving mapping of diagnostic information. Two typical solutions are similar to these which were presented in previous points e.g.:
  auxiliary model: position := position(encoder),
  rules: IF NOT encoder_ok AND flow_max THEN position := 15 AND value := 30,
- Graph-graph configuration. The situation is quite similar to previous point.

6. Example

The root of the *tree of components* is a simple arithmetic circuit (Fig. 7) Reiter, [15].

The circuit consists of three multipliers: $m1$, $m2$, and $m3$; and two adders: $a1$ and $a2$. The values of inputs are $A = 3, B = 2, C = 2, D = 3$, and $E = 3$. The values of outputs are $F = 23$ and $G = 12$. So, there is an inconsistency between the measured value of output F and its predicted value.

For the arithmetic circuit, minimal conflict sets are: $\{a1, m1, m2\}$, $\{a1, a2, m1, m3\}$. Diagnosis (minimal hitting sets) are: $d_0 = \{a1\}, d_1 = \{m1\}, d_2 = \{m2, m3\}, d_3 = \{a2, m2\}$.

Let us consider diagnosis $d_0 = \{a1\}$. Focus function maps observations to attributes of the diagnosis subcomponents $ATR^1_{d_0} = \{X = 6, Y = 6, F = 23\}$.

Subcomponent $a1$ can be described by the model which is presented in Fig. 8. Hierarchical function "distributes" values of attributes to observations at this level: $0X = 0, 1X = 1, 2X = 1, 3X = 0, 0Y = 0, 1Y = 1, 2Y = 1, 3Y = 0, 0F = 1, 1F = 1, 2F = 1, 3F = 0, 4F = 1$.

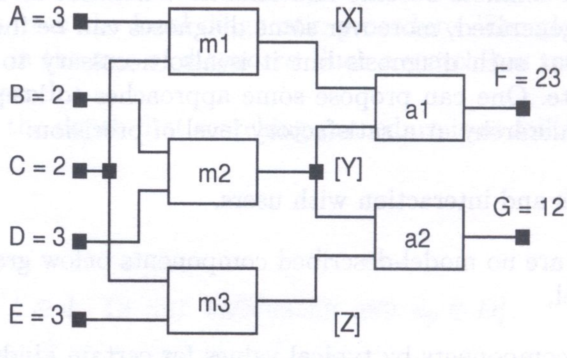

Fig. 7. Arithmetic circuit

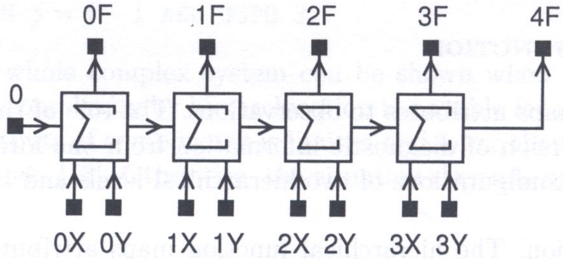

Fig. 8. Full adder

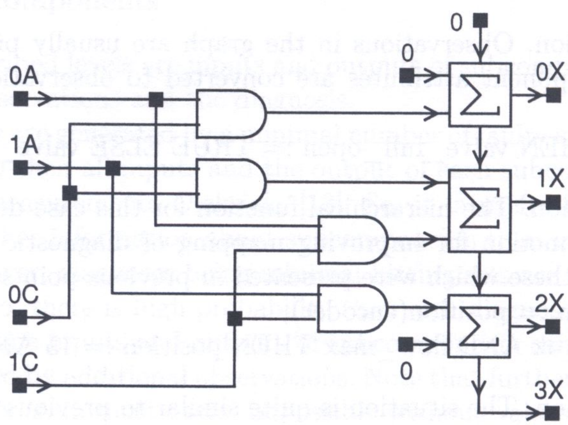

Fig. 9. Multiplier

A minimal diagnosis at this level is $\{\Sigma_0, \Sigma_1\}$. If components included in the diagnosis are *complex components* then diagnosis can be continued at lower level of hierarchy.

Some diagnoses can be limited by the focus function. For example, one from the diagnoses for the arithmetic circuit in Fig. 7 is $\{m1\}$. The value at the output of faulty subcomponent $\{m1\}$ should be equal 17. Now, let us take a look on multiplier $\{m1\}$ at a lower level of hierarchy (Fig. 9).

The model of the multiplier shows that the multiplier is not capable of producing output value equal 17 (15 is maximum). 17 is out of domain of the multiplier output and diagnosis $\{m1\}$ will be rejected.

The *tree of components* for the hierarchical model of the circuit is presented in Fig. 10.

And one more example of limitation which is typical for graph-described components and occurs when any behavior of subcomponents, either normal or abnormal, cannot explain observations.

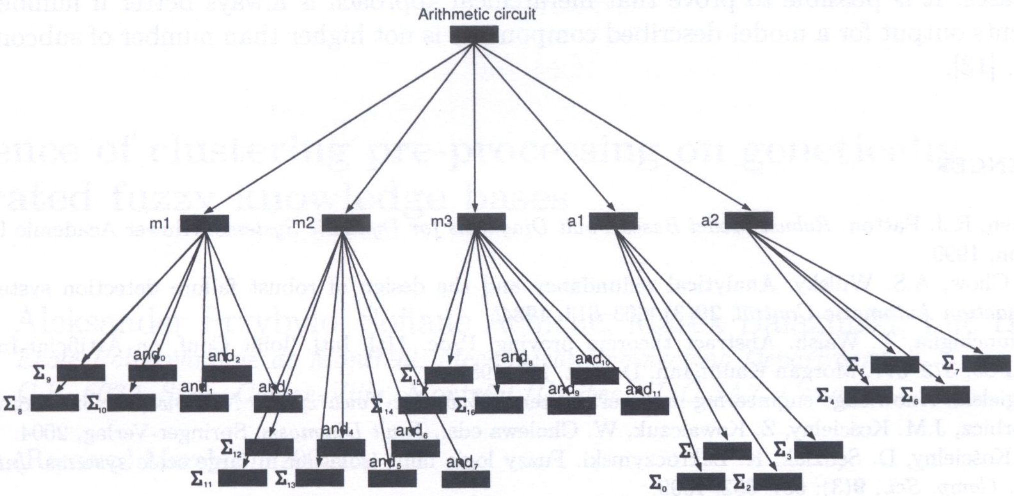

Fig. 10. The tree of component structure for the circuit

In other words, a logical function, which is coupled to the expert graph, cannot be fulfilled for current observations and any values of other parameters.

For example, Fig. 11 presents simple expert graph modeling AND gate. The gate has two inputs *in1(AND) = 0*, *in2(AND) = 0* and one output *output(AND) = 1*. Such values of observation cannot be explained by any value of variable *AB(AND)*.

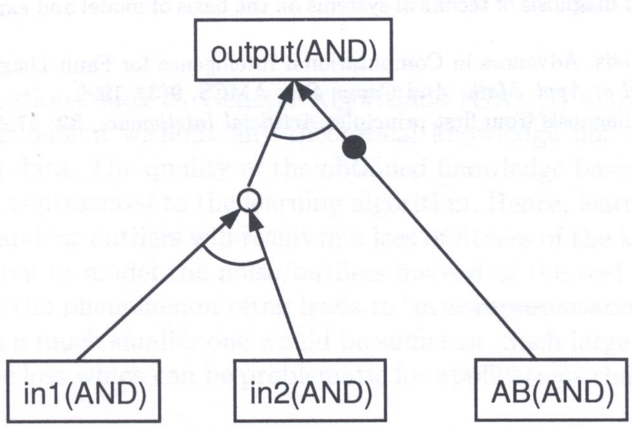

Fig. 11. Causal graph for AND gate

Reasons for such situation are either weakness of the model or intentional elimination of some states from the model.

7. Concluding remarks

This article presented a flexible and easy methodology for hierarchical modeling and diagnosing of complex technical systems with different kinds of diagnostic descriptions for components. The computational cost of the diagnostic procedure depends on models and complexity of the local diagnostic procedures, but hierarchical diagnosis is less costly than single-level diagnosis in most typical cases. It is possible to prove that hierarchical approach is always better if number of subcomponents output for a model-described components is not higher than number of subcomponents Oleksiak, [13].

References

[1] J. Chen, R.J. Patton. *Robust Model Based Fault Diagnosis for Dynamic Systems.* Kluwer Academic Publishers, Boston, 1999.

[2] E.Y. Chow, A.S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transaction Automatic Control,* **29**(3): 603–614, 1984.

[3] F. Giunchiglia, T. Walsh. Abstract theorem proving. Proc. 11th Intl. Joint Conf. on Artificial Intelligence, IJCAI-89, 372–377, Morgan Kaufmann, Detroit, MI, 1989.

[4] J. Jagielski. Knowledge engineering in expert systems. *Lubuskie Towarzystwo Naukowe,* Zielona Góra, 2001.

[5] J. Korbicz, J.M. Kościelny, Z. Kowalczuk, W. Cholewa eds., *Fault Diagnosis.* Springer-Verlag, 2004.

[6] J.M. Kościelny, D. Sędziak, K. Zakroczymski. Fuzzy logic fault isolation in large scale systems. *Int. J. Appl. Math. Comp. Sci.,* **9**(3): 637–652, 1999.

[7] J.M. Kościelny. Diagnosis of automatic industrial processes. AOW EXIT, 2001.

[8] A. Ligęza, P Fuster-Parra. And/or/not casual graphs – a model for diagnostic reasoning. *Applied Mathematics and Computer Science,* **7**(1): 57-95, 1997.

[9] A. Ligęza. Selected methods of knowledge engineering in fault diagnosis. In: J. Korbicz, J.M. Kościelny, Z. Kowalczuk, W. Cholewa, eds., *Fault Diagnosis* (WNT, 581–622, 2002.

[10] W.A. Moczulski, Technical fault diagnosis. Methods of knowledge acquisition. Wydawnictwo Politechniki Śląskiej, Gliwice, 2002.

[11] I. Mozetič, Hierarchical model-based diagnosis. *International Journal of Man-Machine Studies,* **3**(5): 329–362, 1991.

[12] J. Oleksiak, A. Ligęza, Hierarchical diagnosis of technical systems on the basis of model and expert knowledge. *Recent Developments in Artificial Intelligence methods* T. Burczyński, W. Cholewa and W. Moczulski, eds. AI-METH Series, Gliwice, Poland, 199–203, 2004.

[13] J. Oleksiak, Hierarchical diagnosis of technical systems on the basis of model and expert knowledge. PhD Thesis, 2004.

[14] R.J Patton, J. Korbicz eds. Advances in Computational Intelligence for Fault Diagnosis Systems. *Special Issue of International Journal of Appl. Math. And Comp. Sci.,* AMCS, **9**(3), 1999.

[15] R. Reiter. A theory of diagnosis from first principles.*Artificial Intelligence,* **32**: 57–95, 1987.