Fault Diagnosis with Progressive Symptoms Based on Multi-Agent Approach

Oleksandr Sokolov
National Aerospace University
Tchkalov str. 17, Kharkiv 61070
Ukraine
asokolov@xai.edu.ua

Michael Wagenknecht
University of Applied Sciences
Zittau/Goerlitz, IPM
Theodor-Koerner-Allee 16
Zittau 02763, Germany
m.wagenknecht@hs-zigr.de

Ulrike Gocht
University of Applied
Sciences Zittau/Goerlitz, IPM
Theodor-Koerner-Allee 16
Zittau 02763, Germany
u.gocht@hs-zigr.de

Abstract

The paper is devoted to fault diagnosis problems using fuzzy decision making.
We investigate dynamic diagnostic systems which can be represented by symptom-fault rule bases. The main question to be answered is what faults produce observable symptoms in the first moments of their appearance. To solve this task we propose to use a set of agents for making fuzzy hypothesis about symptoms and to solve inverse fuzzy relations.

Keywords: symptom-fault model, agent, fuzzy, diagnosis.

1 Symptom-fault model

We use $F = \{f_l\}_{l=1}^{N_F}$ and $S = \{s_m\}_{m=1}^{N_S}$ to denote the finite sets of all possible faults and symptoms, respectively.

There is also a multi-valued mapping

$$\psi : S \to F \quad (1)$$

that can be presented as a ***binary diagnostic matrix*** like in Table 1 reflecting the influence of relations between elements of sets S and F as numbers from $I = \{0,1\}$.

The propagation of faults to observable symptoms follows in general physical cause-effect relationships.

Figure 1 shows that a fault in general influences events as intermediate steps which

Table 1: Example of a binary diagnostic matrix

S / F	f_1	...	f_{N_F}
s_1	1	...	0
...	0	...	1
s_{N_S}	1	...	1

then influence the measurable or observable symptoms, both by internal physical properties. The fault diagnosis proceeds the reverse way. It has to conclude from the observed symptoms to the faults [1]. This implies the inversion of the causality. We propose to consider these intermediate events between faults and symptoms as a set of agents that are responsible for observing of symptoms and making hypothesis about possible faults [2].

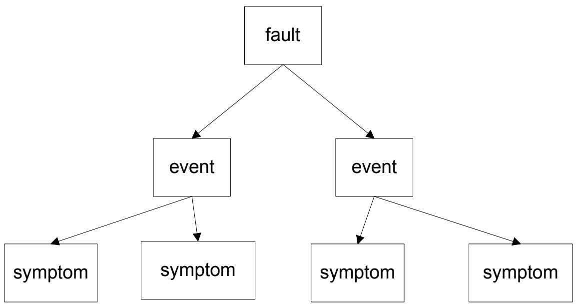

Figure 1: Fault-event-symptom diagram.

Usually the diagnostic methods solve the fault isolation problem using direct mapping 'symptom-fault'.

The main mission of agents in diagnostic task is to perceive the symptoms on first stages of their arising and recognize caused faults.

Additional advantages of using agents are:

- spatially distributed diagnosis,
- activities of events,
- possibility to communicate [3].

We can formulate the diagnosis tasks according to the multi-agent approach [1]. The general goals of fault diagnosis are:
(1) to detect as many faults as possible;
(2) to detect the faults as soon as possible;
(3) to avoid false alarms.

Let the set of agents given by $A = \{a_i\}_{i=1}^{N_A}$, $N_A \le N_F$ and consider a map

$$\xi : A \to F \tag{2}$$

fulfilling the following conditions

$\xi(A) = F$,
$\forall i, j \ne i \in N_A \;\; \xi(a_i) \ne \xi(a_j)$,
There may exist
$i, j \ne i \in N_F \;\; \xi^{-1}(f_i) = \xi^{-1}(f_j)$

It means that each agent may be responsible for several faults but one fault can be recognized by only one agent. The way in which the set F is divided by agents from A is a state of art of the subject diagnostic area. The natural manner is to divide faults according to subsystems of diagnostic objects [5].

2 Multi-agent diagnosis model

First of all we need to give the mechanism of information of agents *A* about observed symptoms *S*. It may be a new mapping $\vartheta : S \to A$ or a general information about faults in form of logical expression obtained from a binary diagnostic matrix. We consider two tasks.
We suppose here that all agents obtain information about faults from mapping (1) (binary diagnostic matrix).

Let $S_O \subseteq S$ be a subset of symptoms observed by some management application. It reduces to the generation of an appropriate logical expression from (1) in Conjunctive Normal Form, i.e. from $\psi|_{S_O} : S \to F$ we can get

$$\sigma = \bigvee_{i=1}^{D(S_O)} \left(\bigwedge_{j \in C_i(S_O)} f_j \right) \tag{3}$$

where $D(S_O)$ is a number of disjunctions in the formula, $C_i(S_O)$ is a set of indexes in the i^{th} conjunction expression.

Formula (3) gives the *minimum number of disjunctions* of faults that cause the set of observed symptoms. It is a necessary condition for getting S_O. Of course, the full set of possible combinations of faults (sufficient condition) σ_{full} that could be generated from (3) is expected to be essentially larger, but this is not realistic in practice.

Now we can formulate the task of ***faults refinement*** in terms of *cooperation logic*. According to [4] cooperation logic is a logic that is to enable reasoning about coalitions in multi-agent systems, and in particular, the *power* which such coalition has.

Coalition logic systems are based upon the notion of a cooperation modality: a unary modal operator, indexed by a set of agents which is used to represent the fact that this set of agents can cooperate so as to make ***true*** the state of affairs given as an argument to the operator.

In Coalition logic, for example, a formula $[1,2](p \wedge q)$ is used to express the fact that the coalition of agents $\{1,2\}$ can cooperate in such a way as to make the formula $(p \wedge q)$ *true*.

The key idea of using coalition logic in multi-agent fault diagnosis is that each agent is assumed to verify a set of faults assigned to this agent by mapping (2). The strategies (choices) available to an agent then correspond to the different possible assignments of truth of falsity to these propositions. On top of that, the ability of a coalition to bring about some state of affairs derived from fault variables that are under the overall control of the coalition.

Within such an approach we can consider the set of faults as a set of propositional variables that has *true* or *false* values.

We now introduce a model of coalition logic as

$$M = \left\langle A, F, Af_1, \ldots, Af_{N_A}, \theta \right\rangle \tag{4}$$

where

- $A = \{a_i\}_{i=1}^{N_A}$ is a finite, non-empty set of agents;
- $F = \{f_l\}_{l=1}^{N_F}$ is a finite, non-empty set of faults (propositional variables);
- $Af_1, \ldots, Af_{N_A}$ is a partition of F among the members of A, with the intended interpretation that Af_i is a subset of F representing the variables (faults) under the control of agent $a_i \in A$;
- $\theta : F \to \{true, false\}$ is a propositional valuation function, which determines the ***initial truth value*** of every propositional variable (fault).

We have also (see Section 1)

$F = \bigcup_{i=1}^{N_A} Af_i$, i.e., every fault is controlled by some agent, and $Af_i \cap Af_j = \varnothing$ for $a_i \neq a_j \in A$. That is, no fault is controlled by more than one agent.

In case of fault diagnosis tasks we suppose that initial truth value is *false* for all faults from F.

Now a *coalition* C is simply a subset of A, i.e. $C \subseteq A$. Moreover, we set $F_C = \bigcup_{a_i \in C} Af_i$.

Given a model (4) and a coalition C in M, a C-*valuation* is a function:

$$\theta_C : F_C \to \{true, false\}.$$

Thus C-*valuation* is a function that assign truth values to just the primitive propositions controlled by the members of the coalition C. In fault diagnosis tasks, a C-*valuation* is simply checking of fault existence for appropriate agents.

Let us denote by $M \oplus \theta_C$ the model which is identical to M except that the values assigned by its valuation function to propositions controlled by members of C are determined by θ_C.

Given a model $M = \langle A, F, Af_1, \ldots, Af_{N_A}, \theta \rangle$ and a formula $\sigma = \bigvee_{i=1}^{D(S_O)} \left(\bigwedge_{j \in C_i(S_O)} f_j \right)$ we write $M \models^d \sigma$ to express that σ is satisfied (or, equivalently, *true*) in M, under the 'direct' semantics. The rules defining the satisfaction relation $\models^d$ are as follows:

- $M \models^d \top$;
- $M \models^d f$ iff $\theta(f) = true$ (where $f \in F$);
- $M \models^d \neg\varphi$ iff $M \not\models^d \varphi$;
- $M \models^d \varphi \vee \psi$ iff $M \models^d \varphi$ or $M \models^d \psi$;
- $M \models^d \Diamond_C \varphi$ iff there exists a C-*valuation* θ_C such that $M \oplus \theta_C \models^d \varphi$.

The diagnostic task can be formulated now as following:

for model $M = \langle A, F, Af_1, \ldots, Af_{N_A}, \theta \rangle$ and formula $\sigma = \bigvee_{i=1}^{D(S_O)} \left(\bigwedge_{j \in C_i(S_O)} f_j \right)$ it is necessary to find a minimal (in special sense) coalition of agents $C \subseteq A$ for that $M \models^d \Diamond_C \sigma$ remains true. The goal of minimization may be

$$\arg\min_C M \models^d \Diamond_C \sigma \qquad (5a)$$

or

$$\arg\min_F M \models^d \Diamond_{C(F)} \sigma. \qquad (5b)$$

for minimum time.

Because task (5b) is more natural in fault diagnosis we try to find the minimal set of faults that cause observed symptom set S_O.

Let us suppose that all faults are equiprobable and equipollent.

Without the multi-agent approach this task is transformed to the task of checking the faults F_C caused the set S_O element-by-element and each step to test formula (3). Let $F_C = \{F_{pos}, F_{neg}, F_{rest}\}$, where F_{pos} is set of faults taking place and caused symptoms S_O; F_{neg} is the set of faults that were checked, but they do not cause symptoms; F_{rest} is a set possible faults for symptoms S_O. If checking time of one fault is t_f then the time in *successive* algorithm is

$$T_S = t_f \cdot card\left(F_{pos} \cup F_{neg}\right).$$

It is easy to see that a *multi-agent* algorithm should fulfill

$$T_{MA} \le T_S .$$

The worst case is $T_{MA} = t_f \cdot \max_{i=1,\dots,N_A} card\left(Af_i\right)$ and we have an $Af_i = \left\{F_{pos} \cup F_{neg}\right\}$ with $T_{MA} = T_S$. The minimal time in multi-agent approach is $T_{MA} = t_f$ when the number of agents is equal to the number of faults. Thus we have $t_f \le T_{MA} \le T_S$.

The main advantage of the multi-agent approach is parallel calculation and interaction between agents to make decisions.

We consider the fault isolation task as a dynamic process in which diagnostic agents communicate with each other and exchange knowledge to solve the diagnostic problem.

For purpose of communication organization we need to set the protocol of agent's speaking.

Each agent makes decisions about the necessity of checking faults according to the current situation. The agent chooses such a fault that could make formula σ true as soon as possible.

All agents obtain a formula σ for testing.

At the first step (unit of time) agents inform each other for which faults they will minimize time. At the next step they check faults simultaneously by prior arrangement and test formula σ. If consensus is not reached (result is false) σ is reduced according to current values of checked faults and the process is repeated.

We suppose all agents to check their faults ***simultaneously*** at one step and at the next step they exchange messages about the current state of their faults. This procedure will be repeated until consensus is reached. We suppose that this procedure to be finite due to the reliable information of observed symptoms and the truthful binary diagnosis matrix.

The proposed algorithm is presented in Figure 2.

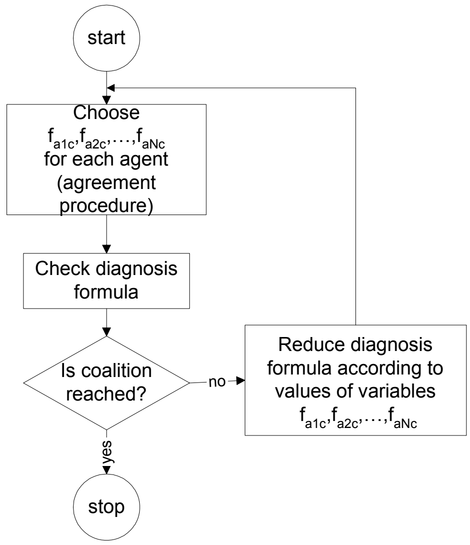

Figure 2: Algorithm of coalition reaching procedure

Let $C = \left\{a_{1_C}, \dots, a_{N_C}\right\}$ be the set of agents responsible for σ. For each agent a_{j_C} there is the set of corresponding faults $Af_{j_C} = \left\{f_{1_{jC}}, \dots, f_{N_{jC}}\right\} \subset F$.

We use the following heuristic recurrent procedure for choosing the set of testing faults from $F_C = \left\{Af_{1_C}, \dots Af_{N_C}\right\}$.

Let $\sigma_0 = \bigvee_{i=1}^{D_0} \left(\bigwedge_{j \in C_i^0} f_j \right)$ be an initial formula (3) where $D_0 = D(S_O)$ is the number of disjunctions in the formula; $C_i^0 = C_i(S_O)$ is the set of indexes in the i^{th} conjunction expression. Let $F_0 = \left\{F_{1_C}^0, \dots F_{N_C}^0\right\} = F_C$ be the set of possible faults that caused the observed symptoms distributed on agents from C. Further we propose the procedure that performs by each agent Ag from $C = \left\{a_{1_C}, \dots, a_{N_C}\right\}$ at the k^{th} step simultaneously.

We call this an α**-procedure**.

1. Refresh

$$\sigma_k = \bigvee_{i=1}^{D_k} \left(\bigwedge_{j \in C_i^k} f_j \right)$$

according to the result of faults checking at the previous step and abbreviate by cancellation excessive variables and false conjunctions (executed one time at each step).

2. ***If*** $F_{Ag}^{k} \neq \emptyset$ ***then*** choose one fault f_{Ag} from F_{Ag}^{k} that belongs to the conjunction in (3) with minimal length $\min card\left(C_i^k\right)$. If there are several conjuctions with the same length we apply a random choice. Set $F_{Ag}^{k+1} = F_{Ag}^{k} \setminus f_{Ag}$ ***else*** *stop* testing process for the current agent.
3. Check f_{Ag} and set $f_{Ag} = \{true, false\}$.
4. Set global state $k = k+1$.
5. Goto 1

The proposed algorithm is converging due to the reliable set of symptoms, the truthful binary diagnosis matrix and the step-by-step reducing of the set of possible faults.
It is possible to improve the procedure not considering all agents simultaneously, but one after another. In cooperation logic such a model is called β-coalition contrary to the α-coalition for the case described above [5]. The β-coalition foresees the conditional coalition when the current decision strongly depends on the previous one.
The β **-procedure** is as follows.

1. Determine
$$\sigma_k = \bigvee_{i=1}^{D_k} \left(\bigwedge_{j \in C_i^k} f_j \right)$$
from the previous step or after modification at the current step.
2. ***If*** $F_{Ag}^{k} \neq \emptyset$ ***then*** choose one fault f_{Ag} from the set F_{Ag}^{k} that belongs to the conjunciton in (3) with minimal length $\min card\left(C_i^k\right)$. Set $F_{Ag}^{k+1} = F_{Ag}^{k} \setminus f_{Ag}$ ***else*** *stop* testing the process for the current agent.
3. ***If*** $\min card\left(C_i^k\right) = 1$ and $F_{Ag}^{k+1} = \emptyset$ ***then*** set $f_{Ag} = true$ without checking.
else check f_{Ag} and set the appropriate value $f_{Ag} = \{true, false\}$.
4. Refresh
$$\sigma_k^{\bmod_{Ag}} = \bigvee_{i=1}^{D_k} \left(\bigwedge_{j \in C_i^k} f_j \right)$$
according to the value $f_{Ag} = \{true, false\}$ and abbreviate by cancelling excessive variables and false conjunctions.
5. Send to the next agent (from $C = \{a_{1_C}, \ldots, a_{N_C}\}$) the modified formula $\sigma_k^{\bmod_{Ag}}$. If this agent is last in the list then set $\sigma_{k+1} = \sigma_k^{\bmod_{Ag}}$.
6. After testing (4.9) by the remaining agents set global state $k = k+1$.
7. Goto 1

Step 5 demands to set the schedule of agents for sending information. This schedule may be organized by

- a list in $C = \{a_{1_C}, \ldots, a_{N_C}\}$;
- randomly;
- in inverted range of cardinality of elements in $F_k = \{F^k{}_{1_C}, \ldots F^k{}_{N_C}\}$.

Remark.
Progressive symptoms mean that we use fuzzy logic for their evaluation and calculate the possible set of faults using fuzzy inference.

Conclusion
Due to the distributional character of fault diagnosis and necessity of taking into account different uncertainties during the diagnosis process the multi-agent methodology has been proposed. We showed that the principle of assigning the agent to subsystems in this case is also useful. Then the local knowledge base of agents and rules for communications can precise the possible diagnosis in addition to the ordinary binary or fuzzy diagnostic matrix.

As a result of this investigation we could propose engineering methods for applying multi-agent methods to fault diagnosis problems.

Acknowledgement

This research has partially been supported by the Saxon Ministry of Sciences and Art (Germany).

References

[1] R. Isermann. Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance. Springer, 2006.

[2] Soo Young Eo, Tae Suk Chang, Dongil Shin, En Sup Yoon. Cooperative problem solving in diagnostic agents for chemical processes. Computer & Chemical Engineering 24 (2000) 729 – 734.

[3] M. Wooldridge. An Introduction to Multi-Agent Systems. John Wiley&Sons, 2001.

[4] W. Hoek, M. Wooldridge. On the logic of cooperation and propositional control. Artificial intelligence 164 (2005) 81-119.

[5] B.D. Pomeroy, H.A. Spang, M.E. Dausch. Event-based architecture for diagnosis in control advisory systems. Artificial Intelligence in Engineering 5 (1990) 174-181.