Decentralized Diagnosis in Heavy Duty Vehicles

Jonas Biteus*, Mattias Jensen† and Mattias Nyberg†
* Dept. of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
E-mail: biteus@isy.liu.se
† Software and Diagnostics, Engine Development
SCANIA AB, SE-151 87 Södertälje, Sweden
E-mail:{mathias.jenssen,mattias.nyberg}@scania.com

***Abstract*— Fault diagnosis is important for automotive vehicles, due to economic reasons, such as efficient repair and fault prevention, and legislative reasons, mainly safety and pollution.**

Embedded systems in vehicles includes a large number of electronic control units, that is connected to each other via an electronic network. Many of the current diagnostic systems use pre-compiled diagnostic tests, and fault-logs that stores the results from the tests. To improve the diagnostic system, fault localization in addition to the existing fault detection is wanted.

Since there are limitations in processing power, memory, and network capacity, an algorithm is searched for that uses stated diagnoses in the control units to find the diagnoses for the complete system. Such an algorithm is presented and exemplified in the article.

The embedded system used in a Scania heavy duty vehicle, has been used as a case study to find limitations in the embedded system, and realistic requirements on the algorithm.

I. Introduction

In most modern automotive vehicles, several *electronic control units* (ECU) communicates over an electronic network. Each ECU is usually connected to one or several *components*, e.g. sensors and actuators, and to make sure that the components are operating correctly, they are *monitored* by the ECUs. Often, *precompiled diagnostic tests*, which can be simple or complex, are used to perform the monitoring.

If an ECU, with the use of tests, finds that one or several components are showing abnormal behaviors, one or several *diagnostic trouble codes* (DTC) are created. These DTCs are read out by technicians, and will assist them in locating the faulty component or components. When reading the DTCs, it might be difficult to localize the faulty component, or components, among all the components that are included in the DTCs. The reasons for this are that each DTC can involve multiple components, and that the sheer volume of DTCs might be overwhelming for the technician. Due to this difficulty, the reparation times might be prolonged, and non-faulty components might unnecessary be replaced.

Therefore, it is preferable to present a list of the possibly faulty components that might have caused these DTCs, i.e. the *diagnoses* that explain the DTCs should be found.

The diagnoses that are calculated for a single ECU are here denoted the *local diagnoses*. When local diagnoses have been calculated for an ECU, these describes the possible faulty components that are somehow involved with this ECU's operation. In the set of local diagnoses, there are some diagnoses that are more likely than the other diagnoses, e.g. a diagnosis that only includes a sub-set of the components in another diagnosis. If all the local diagnoses in all the ECUs were merged together, the result would be the *global diagnoses*, i.e. the diagnoses for the complete system.

After such a simple merge of the local diagnoses, one could find that some of the local diagnoses, that from a local point of view seemed to be most likely, *is in fact not at all most likely considering the complete system*. How to calculate the more likely global diagnoses from the local diagnoses is the topic of this paper.

Notice that an alternative when searching for the global diagnoses, would be to start all over again from the DTCs, however, since the local diagnoses already have been computed, it might be preferable to use these.

An *algorithm* is presented that uses the individual ECUs' computation power to calculate the more likely global diagnoses from local diagnoses.

A. A Typical Embedded System

Many vehicles, including Scania's heavy-duty vehicles, have a *controller area network* (CAN) which connects several ECUs to each other. Each of the ECUs, is further connected to sensors and actuators, and both sensor values and control signals can be shared with the other ECUs over the network. An example of an ECU, is the engine management system, which is connected to sensors and actuators related to the engine. In Fig. 1, a configuration of the current Scania embedded system is shown. It includes three separate CAN buses, red, yellow, and green, which are connected to the coordinator ECU. The coordinator acts as a router, making sure that no messages are forwarded to any other bus unless it is necessary. There are between 20 and 30 ECUs in the system, depending on truck's type and outfit, and between 4 and 110 components are connected to each ECU. The ECUs' CPUs have a clocking speed of 8 to 64 MHz, and a memory capacity of 4 to 150 kB.

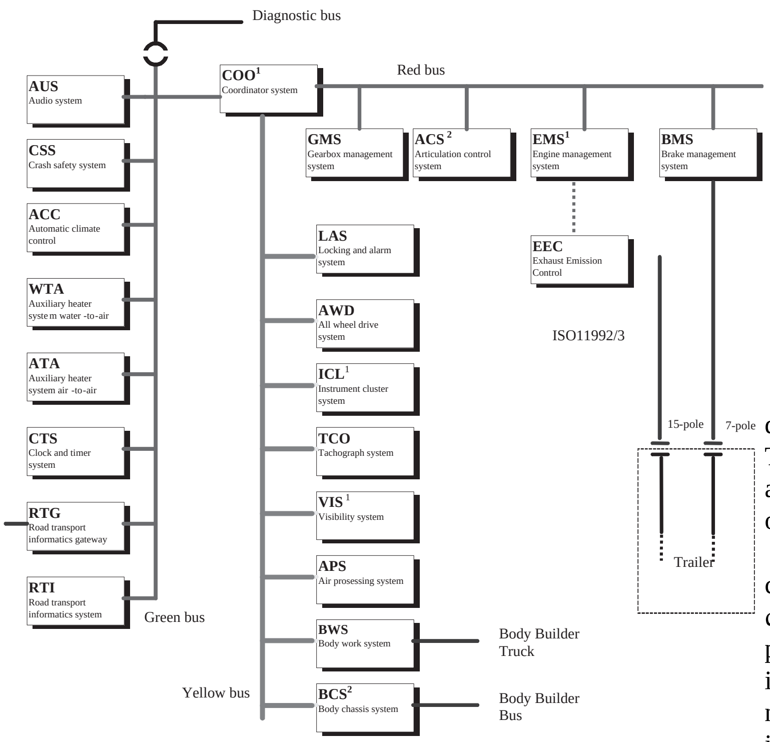

Fig. 1. The embedded system in current Scania heavy-duty vehicles.

B. A Typical Diagnostic System

The current Scania diagnostic system consists of precompiled diagnostic tests. When a test indicates some fault, a diagnostic trouble code (DTC) is generated. There are between 10 and 1000 tests in each ECU. No *explicit* fault isolation is performed in the current system, neither in or between the ECUs. Further, there are no explicit communication between the ECUs, regarding the results of the tests.

In Fig. 2, a typical layout of ECUs and components are shown. The tests involves the components c_i that is connected with solid lines. Some calculated value used in a test in agent A_1 which involves component c_2 and c_4 is transmitted over the network from agent A_1 to A_2.

C. Related Work

Diagnosis for embedded systems can be centralized, decentralized or distributed. Most research has been aimed at the centralized problem, where a single process collects relevant data from the system and states global diagnoses [1]. In contrast to the centralized system, in a fully distributed system, there is no central process at all. The distributed processes communicates instead directly with each other to form the global diagnoses [2], [4].

In a decentralized system, there is some central diagnostic process that directs the diagnostic work, while local processes are doing some part of the diagnostic work [3].

II. System Description and Diagnosis Representation

The diagnostic system involves a set of agents, $\mathcal{A}$, and a set of components, $\mathcal{C}$, which are the set of objects that can be

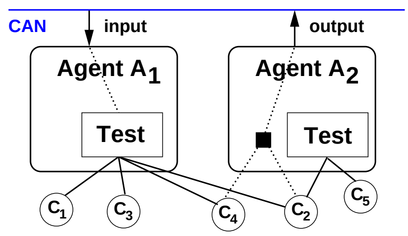

Fig. 2. A typical ECU, component and test layout.

diagnosed, by one or several agents, for abnormal behavior. The agents are connected to each other via a network, where an output signal in an agent is linked to input signals in one or several other agents.

Example 1: In Fig. 2, a typical layout of agents and components is shown. The tests involves the components c_i connected with solid lines. Components c_2 and c_4 are physically connected to agent A_2, and some calculated value involving components c_2 and c_4 is transmitted over the network from agent A_2 to A_1, indicated by the dotted lines in the figure. ◇

In a vehicle, the agents are the software programs that are implemented in the ECUs. The components are the sensors, actuators, pipes, etc., which are monitored by the diagnostic systems. Output signals are mostly values from sensors, actuator signals, or calculated values, that are made available to the other agents over the CAN.

Each component, $c \in \mathcal{C}$, has a general fault mode and a no-fault mode. Since the general fault mode does not have a model, the notation used in for example GDE is here employed [6].

A diagnosis for a set of components $\mathcal{C}' \subseteq \mathcal{C}$ is

$$D \subseteq \mathcal{C}'$$

It states that the components $c \in D$ are in the faulty mode, while the remaining components $\mathcal{C}' \backslash D$ might or might not be in the faulty mode.

A *local diagnosis* is a diagnosis stated by an agent, it states the modes of some sub-set of components, $\mathcal{C}'$, which are somehow involved in the operation of the agent. The set of local diagnoses for agent A_i is $\mathbb{D}_i$.

A *global diagnosis* is a diagnosis that can state the mode of all components in the system, i.e. $\mathcal{C}' = \mathcal{C}$. The set of global diagnoses is denoted $\mathcal{D}$.

Due to the combinatorial explosion when searching for global diagnoses, it is preferable to only consider some subset of global diagnoses with properties that are specifically interesting. One such set is the *minimal-cardinality* (mc) *diagnoses*

$$\mathcal{D}^{mc} = \{D_j \,\big|\, |D_j| = \min_{D_i \in \mathcal{D}} |D_i|, D_j \in \mathcal{D}\}$$

for a set of diagnoses $\mathcal{D}$. The set of local minimal cardinality diagnoses for agent A_i is denoted $\mathbb{D}^{mc,i}$, and the set of global minimal cardinality diagnoses is denoted $\mathcal{D}^{mc}$.

If all components have the same probability to fail, the most likely global diagnoses will simply be the global minimal cardinality diagnoses. This is one such set of more likely global diagnoses, that according to Section I is wanted.

In Section I, the possibility to merge different sets of local diagnoses was discussed. The *merging* of two sets of diagnoses, $\mathbb{D}_i$ and $\mathbb{D}_j$, is defined as $\mathbb{D}_i \times_\cup \mathbb{D}_j \triangleq \{D_i \cup D_j \,|\, D_i \in \mathbb{D}_i, D_j \in \mathbb{D}_j\}$. By merging all local diagnoses, the global diagnoses can be formed

$$\mathcal{D} = \times_{\cup i} \mathbb{D}_i$$

Example 2: [Merge of local diagnoses] Consider the local diagnoses $\mathbb{D}_1 = \{\{A, B\}, \{C\}\}$ and $\mathbb{D}_2 = \{\{B\}\}$. The global diagnoses $\mathcal{D} = \mathbb{D}_1 \times_\cup \mathbb{D}_2 = \{\{A, B\}, \{C, B\}\}$. ◇

It is however not the case that a merge of the local minimal cardinality diagnoses results in the global minimal cardinality diagnoses, i.e.

$$\times_{\cup \atop i} \mathbb{D}_i^{mc} \neq \mathcal{D}^{mc}$$

for some sets of local minimal cardinality diagnoses.

Example 3: [Merge of $\mathbb{D}_i^{mc}$] Consider the example above, where $\mathbb{D}_1^{mc} = \{\{C\}\}$ and $\mathbb{D}_2^{mc} = \{\{B\}\}$. The merge gives, $\mathbb{D}_1^{mc} \times_\cup \mathbb{D}_2^{mc} = \{\{C, B\}\}$, while $\mathcal{D}^{mc} = \{\{A, B\}, \{C, B\}\}$, i.e. $\{A, B\}$ was not included in the merge of the local minimal cardinality diagnoses. ◇

A. Global Diagnosis Representation

The global diagnoses can be represented in several different ways. When considering systems, such as a vehicle, the diagnoses should be used by a technician for repair. In this case, it might be preferable to present the global diagnoses as a conjunction of smaller disjoint parts of diagnoses. The global diagnoses will then simply be a merge of all the disjoint parts, which is more compactly, and easily, represented by the parts themselves. From the technicians point of view, the parts will be smaller, and therefore more easily understood, than the complete set of global diagnoses.

If the local diagnoses were disjoint, one such conjunction would be available directly from the local diagnoses. This might however not be the case, because the different local diagnoses might include complex relations between diagnoses, i.e. they are not disjoint. Another way is to merge the local diagnoses from a sub-set of agents, so that each such set of merged diagnoses is disjoint from the others. These sets of diagnoses are here denoted *module diagnoses*, $\mathcal{D}^{mod} \triangleq \times_{\cup A_i \in \bar{\mathcal{A}}} \mathbb{D}_i$ for some set of agents $\bar{\mathcal{A}} \subseteq \mathcal{A}$. For two different module diagnoses, $\bar{A}_i \cap \bar{A}_j = \emptyset$ and $D_i \cap D_j = \emptyset$, where $D_i \in \mathcal{D}^{mod,i}$ and $D_j \in \mathcal{D}^{mod,j}$.

Example 4: [Module diagnoses] If $\mathbb{D}_1 = \{\{A, B\}\}$, $\mathbb{D}_2 = \{\{B, C\}\}$, and $\mathbb{D}_3 = \{\{E\}\}$, then for the sub-sets of agents $\bar{\mathcal{A}}_1 = \{A_1, A_2\}$ and $\bar{\mathcal{A}}_2 = \{A_3\}$, it follows that $\mathcal{D}^{mod,1} = \{\{A, B, C\}\}$ and $\mathcal{D}^{mod,2} = \{\{E\}\}$. ◇

The m:th set of *module minimal cardinality diagnoses* (MGMCD) is denoted $\mathcal{D}^{mod,mc,m}$. In contrast to the case with merged local cardinality diagnoses, when considering MGMCDs

$$\times_{\cup \atop i} \mathcal{D}^{mod,mc,i} = \mathcal{D}^{mc}$$

since the different MGMCDs are disjoint.

Example 5: [MGMCDs] Consider Fig. 3 where five agents states local diagnoses in a truck. The local diagnoses are represented by the circles and states the modes of the components inside the circles. Three sets of MGMCDs could be found. ◇

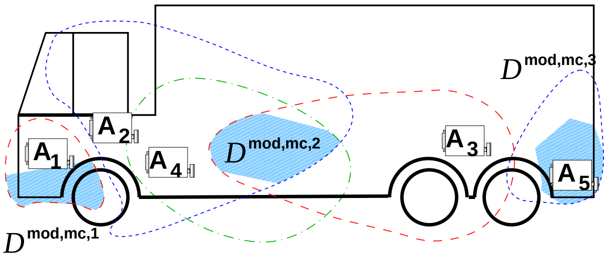

Fig. 3. Agents, local diagnoses, and MGMCDs.

III. Minimal-Cardinality Diagnoses

The principle for the algorithm, is that each agent has created *minimal local diagnoses* using for example the algorithms presented in [1]. These local diagnoses are then merged together to form the MGMCDs.

A. Minimal-Cardinality Diagnoses – Algorithm 1

The algorithm, which is described in Algorithm 1, consists of two main parts. Firstly, is the sets of agents whose local diagnoses are guaranteed to be merged into MGMCDs found. Thereafter, the local diagnoses are iteratively merged with Algorithm 2 into sets of MGMCDs.

The main idea is to start with a low $\mathcal{L}$, then if no partial diagnosis with cardinality less than or equal $\mathcal{L}$ is found, i.e. $N = \emptyset$, then $\mathcal{L}$ is raised and a new search begins. If local diagnoses from all agents have been merged and a partial diagnosis have been found, i.e. $N^{R_n} \neq \emptyset$, this will be the set of MGMCDs. A first approximation of the lower bound, $\mathcal{L}$, is that the MGMCDs must be at least as large as the *largest* minimal cardinality local diagnosis, considering all agents in R.

The algorithm is *decentralized* where Algorithm 1 is calculated in some central agent, while the computation and memory intensive Algorithm 2 is performed in the local agents. In the algorithms, an *ordered* set is represented by $(\cdot)$ and an *unordered* by $\{\cdot\}$.

Example 6: [Algorithm 1] Consider Fig. 4 where the MGMCDs for the three agents should be found. Some central unit decides that the local diagnoses should be merged in order A_3, A_4 then A_2. It sends the UpdateAgent command to agent A_3, who calculates N^{A_3}. Thereafter are the UpdateAgent command sent to A_4 who collects N^{A_3} and calculates N^{A_4}. The end result is $\mathcal{D}^{mod,mc} = N^{A_2}$. ◇

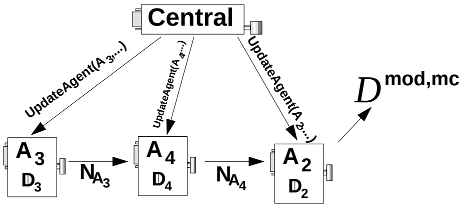

Fig. 4. Information flow.

Algorithm 1 module global minimal-cardinality diagnoses

Require: All local diagnoses, $\mathbb{D}$, for all agents $\mathcal{A}$.
Ensure: All MGMCDs, $\mathcal{D}^{mod,mc}$.

$E := \{(A_i, c) \mid A_i \in V^{\mathcal{A}}, c \in D_j, D_j \in \bar{\mathbb{D}}_i\}$
$\mathbb{G} := (\mathcal{A}, \mathcal{C}, E)$ [*Bipartite graph.*]
Find subgraphs $G \in \mathbb{G}$.
for all $G_i \in \mathbb{G}$, where $G_i = (V_i^{\mathcal{A}}, V_i^{\mathcal{C}}, E_i)$, and $V_i^{\mathcal{C}} \neq \emptyset$. The set $R = V_i^{\mathcal{A}}$, where $R = (R_1, \ldots, R_n)$ **do**
  $\mathcal{L} := \max_{A_i \in R} \min_{D_j \in \mathbb{D}_i} |D_j|$
  $i := 1$
  while $N^{R_n} = \emptyset$ **do**
    $\mathcal{L}^{new} :=$ UpdateAgent$(R_i, R_{i-1}, \mathcal{L})$
    N^{R_i} includes the new diagnoses and is stored in R_i.
    $i := i + 1$
    if $\mathcal{L}^{new} > \mathcal{L}$ **then**
      $i := 1$, $\mathcal{L} := \mathcal{L}^{new}$
    end if
  end while
  $\mathcal{D}^{mod,mc,k} = N^{R_n}$, where R is the k:th $R \in \mathbb{R}$.
end for

B. Update Agent – Algorithm 2

When executing algorithm Algorithm 2 with command UpdateAgent$(R_i, R_{i-1}, \mathcal{L})$, a set of new diagnoses are formed and stored in R_i. These new diagnoses are the diagnoses that are formed by merging the local diagnoses in the part of R up to agent R_i, that have a cardinality less than or equal the given lower limit $\mathcal{L}$. Further, only the diagnoses that have not been considered in previous evaluations are stored in R_i.

In the algorithm, l is the number of times that the current agent have been called with command UpdateAgent$(\cdot)$, e.g. the first time that an agent is called with UpdateAgent, then $l = 1$. The main parts of the algorithm are the merge of the *old and new global diagnoses* for $\{R_1, \ldots, R_{i-1}\}$ with the *new local diagnoses* (W_1), and the merge of the *new global diagnoses* for $\{R_1, \ldots, R_{i-1}\}$ with the *old local diagnoses* (W_2).

Example 7: [Algorithm 2 and relevant part of Algorithm 1] Consider the system in Fig. 4. In includes three agent whose local diagnoses are

$$\mathbb{D}_2 = \{\{B, C\}, \{D\}\} \qquad \mathbb{D}_3 = \{\{B\}, \{C, D, E, F, G\}\}$$
$$\mathbb{D}_4 = \{\{C\}, \{D\}\}.$$

The sort order $R = (A_3, A_4, A_2)$. With $\mathcal{L} = 1$, the first agent, A_3 finds new diagnoses with cardinality $\leq \mathcal{L}$, which give $N^{A_3} = \{\{B\}\}$. and is then finished. Agent A_4 collects N^{A_3} over the network and updates its information,

$$W^{A_4} := \{\{B, C\}, \{B, D\}\} \qquad N^{A_4} := \emptyset.$$

Since $N^{A_4} = \emptyset$ and this is the first time that A_4 have been called, i.e. $l = 1$, $\mathcal{L}$ is increased to 2. Since a new $\mathcal{L}$ was returned the first agent is called again. Agent A_3 updates its information which give $N^{A_3} := \{\{\}\}$. A_4 collects the new information and find that, $N^{A_4} := \{\{B, C\}, \{B, D\}\}$. A_2 takes over and calculates

$$W^{A_2} := \{\{B, C\}, \{B, C, D\}, \{B, D, C\}, \{B, D\}\}$$
$$N^{A_2} := \{\{B, C\}, \{B, D\}\}.$$

Since $N^{R_n} = N^{A_2} \neq \emptyset$, the iterations ends and the set of MGMCDs, $\mathcal{D}^{mod} = N^{A_2}$. $\diamond$

Algorithm 2 UpdateAgent

Require: $\mathbb{D}_1$ for A_1, where $A_1 = R_i$ and $\mathbb{D}_2$ for A_2, where $A_2 = R_{i-1}$. The number of evaluations, including this, of UpdateAgent for agent A_1 is l.
Ensure: $\mathcal{D}^{mod}$ for agents $\{R_1, \ldots, R_i\}$ with cardinality less then or equal $\mathcal{L}$. A new $\mathcal{L}$.

$E := \{D_j \mid |D_j| \leq \mathcal{L}, D_j \in \mathbb{D}_{R_i}\}$
$D := \{D_j \mid |D_j| > \mathcal{L}, D_j \in \mathbb{D}_{R_i}\}$
if $i = 1$ **then** [*The first* R]
  $N := E$
else if $i > 1$ **then**
  $T_l := N^{R_{i-1}}$, where $N^{R_{i-1}}$ is N in agent R_{i-1}.
  $W_1 := \bigcup_{j=1}^{l} T_j \times_\cup E$
  $W_2 := T_l \times_\cup F$
  $W := W_1 \cup W_2 \cup M$
  $F := F \cup E$
  $N := \{W_j \mid |W_j| \leq \mathcal{L}\}$
  $M := W \backslash N$
  if $N = \emptyset \land l = 1$ **then**
    $\mathcal{L} := \min_j |W_j|$ [*New lower bound.*]
  end if
end if
Saved for later use are $\{T_1, \ldots, T_l\}$, D, F, and M. New global diagnoses N is saved for use by agent R_{i+1}.

C. Simulations

The algorithms presented in this paper have not yet been implemented in any vehicle. The reason for this is that the embedded system is very complex, and that the different ECUs are constructed by different companies and can therefore not be easily modified.

Instead, a hypothetical model of an embedded system has been constructed, inspired by the existing system described in Section I-A. The model includes three different buses and 20 ECUs. The components have been divided into three parts, where 65 % are local, 11 % shared within each bus, and 3 % shared within the whole system. Each ECU has about 50 to 200 tests, that together monitor the ECU's components. Both the connections to the shared components, and the tests, are

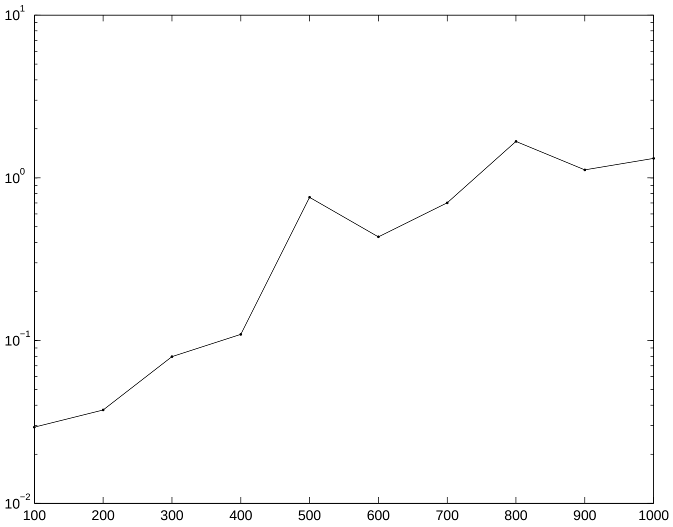

Fig. 5. Calculation times with increasing number of components.

picked by random. Further, between 1 and 5 random faults have been inserted into the model. This results in DTCs from which local diagnoses have been computed.

After this have the algorithm presented in this paper been used to find the MGMCDs. Mean values of the computation times for the algorithm can be seen in Fig. 5, where different number of components have been included in the system.

D. Correctness of the Algorithms

The result of Algorithm 1 is given by the following theorem.

Theorem 1. If all local diagnoses $\mathbb{D}_i$ for all agents $A_i \in \mathcal{A}$ are available, then the result of Algorithm 1 is

$$\mathcal{D}^{mod,mc,k} \tag{1}$$

for all k. ◇

The proof needs two lemmas that shows the correctness of Algorithm 2 and parts of Algorithm 1. The first lemma states that diagnoses of a certain cardinality are found, and that if no diagnoses was found, $\mathcal{L}^{new}$ will have a specific value.

Lemma 2. Given $R = \{R_1, \ldots, R_k\}$*, and a lower limit* $\mathcal{L}^l$*. For* $R_k \in R$*, after computing* $\mathcal{L}^{new} = \text{UpdateAgent}(R_k, R_{k-1}, \mathcal{L})$*, then*

$$N^{R_k} = \{D_j \mid \mathcal{L}^{l-1} < |D_j| \leq \mathcal{L}^l, D_j \in \mathcal{D}\}, \tag{2}$$

where $\mathcal{L}^{l-1}$ *is* $\mathcal{L}$ *from the preceding call to* UpdateAgent*, or if* $l-1 = 0$ *then it is* $\mathcal{L}^{initiation}$*. Further is* $\mathcal{D} = \times_{\cup R_i \in R} \mathbb{D}_i$*. If* $N^{R_k} = \emptyset$ *and* $l = 1$ *for agent* R_k*, then*

$$\mathcal{L}^{new} = \min_j \{|D_j| \mid \mathcal{L} < |D_j|, D_j \in \mathcal{D}\}, \tag{3}$$

otherwise is $\mathcal{L}^{new} = \mathcal{L}^l$. ◇

Proof: Consider first the case where $\Delta\mathcal{L} = 1$. If $k = 1$, i.e. $R = R_1$, then R_1 will have

$$N^{R_1} := E = \{D_j \mid |D_j| = \mathcal{L}, D_j \in \mathcal{D}\}$$

where $\mathcal{D} = \mathbb{D}_{R_1}$. It follows by direct use of the algorithm.

TABLE I

SETS OF DIAGNOSES

$\lvert D_i\rvert, D_i \in \mathbb{D}_{R_k}$	$\lvert D_i\rvert, D_i \in \times_{\cup R_i \in R} \mathbb{D}_i$		
	$< \mathcal{L}$	$\mathcal{L}$	$> \mathcal{L}$
$< \mathcal{L}$	M	W_2	?
$\mathcal{L}$	W_1	W_1	?
$> \mathcal{L}$	?	?	?

For $k > 1$ let $\bar{R} = \{R_1, \ldots, R_{k-1}\}$ and consider agent R_k. Assume that N for agent R_{k-1} is the correct diagnoses with cardinality $\mathcal{L}$ consistent with all diagnoses from agents $\bar{R}$. Further that $T_1, \ldots, T_{k-1}$ are correct.

The following information is stored from previous runs

$$T_k := \{D_j \mid D_j \in \times_{\cup_{R_i \in \bar{R}}} \mathbb{D}_i, |D_j| = \mathcal{L} - l + j, \\ D_{R_i} \in \mathbb{D}_i, |D_{R_i}| \leq \mathcal{L} - l + j\}$$

for $k \in \{1, \ldots, l-1\}$,

$$M := \{D_j \mid D_j \in \times_{\cup_{R_i \in R}} \mathbb{D}_i, |D_j| \geq \mathcal{L}, D_{R_i} \in \mathbb{D}_{R_i}, |D_{R_i}| < \mathcal{L}\} \tag{4}$$

$$F := \{D_j \mid |D_j| < \mathcal{L}, D_j \in \mathbb{D}_{R_k}\} \tag{5}$$

The algorithm give

$$E := \{D_j \mid |D_j| = \mathcal{L}, D_j \in \mathbb{D}_{R_k}\} \tag{6}$$

$$T_l := \{D_j \mid D_j \in \times_{\cup_{R_i \in \bar{R}}} \mathbb{D}_i, |D_j| = \mathcal{L}, D_{R_i} \in \mathbb{D}_i, |D_{R_i}| \leq \mathcal{L}\} \tag{7}$$

The diagnoses

$$D \in \times_{\cup_{R_i \in R}} \mathbb{D}_i$$

are divided into different sets with respect to the cardinality of the diagnoses from R and $\bar{R}$ respectively. In Table I is it shown what parts of the diagnoses that is found in M, W_1, and W_2. M is found from (4). W_2 from (5) and (7). W_1 is calculated from (6) and the union of the T_j:s, where

$$\bigcup_{j=1}^{l} T_j := \{D_j \mid D_j \in \times_{\cup_{R_i \in \bar{R}}} \mathbb{D}_i, |D_j| \leq \mathcal{L}, D_{R_i} \in \mathbb{D}_i, |D_{R_i}| \leq \mathcal{L}\}. \tag{8}$$

From this follows that the new diagnoses

$$N^{R_k} := \{W_j \mid |W_j| \leq \mathcal{L}, W_j \in W\}$$

where $W = W_1 \cup W_2 \cup M$. It must be shown that there is no diagnoses in (2) that has not yet been included in N^{R_k}. The above showed that there is no diagnosis that comes from this or previous calculations that is not included.

The upcoming iterations will give

$$D_{l+1} := \{D_j \mid |D_j| > \mathcal{L}, D_j \in \mathbb{D}_{R_k}\}$$

$$T_{l+1} := N^{R_{k-1}} \\ = \{D_j \mid D_j \in \times_{\cup_{R_i \in \bar{R}}} \mathbb{D}_i, |D_j| > \mathcal{L}, D_{R_i} \in \mathbb{D}_i, |D_{R_i}| \leq \mathcal{L} + 1\}$$

$$E := \{D_j \mid |D_j| > \mathcal{L}, D_j \in \mathbb{D}_{R_k}\}$$

and thereby follows that for iteration $l+1$,

$$W^{l+1} \subseteq \{D_j \,\big|\, |D_j| > \mathcal{L}, D_j \in \mathcal{D}, \mathcal{D} = \times_{\underset{i}{\cup}} \mathbb{D}_i\}$$

then there is no D_j with $|D_j| \le \mathcal{L}$ where $D_j \in \times_{\cup R_i \in R} \mathbb{D}_i$, and for all remaining iterations this will also hold. All diagnoses was therefore included in the set N, i.e.

$$N^{R_k} = \{D_j \,|\, D_j \in \times_{\underset{R_i \in R}{\cup}} \mathbb{D}_i, |D_j| \le \mathcal{L}\}$$

Since $W_1 \ge \mathcal{L}$, $W_2 \ge \mathcal{L}$ and $M > \mathcal{L}^{l-1}$

$$N^{R_k} = \{D_j \,|\, D_j \in \times_{\underset{R_i \in R}{\cup}} \mathbb{D}_i, \mathcal{L}^{l-1} < |D_j| \le \mathcal{L}\}.$$

N^{R_k} was shown given that the assumption about N for agent R_{k-1}, and $T_1, \ldots, T_{k-1}$ are correct. Now, N^{R_1} was above shown to be correct, and therefore the assumption holds by induction.

If $N^{R_k} = \emptyset$

$$\begin{aligned} M &= W \backslash N^{R_k} = W \\ &= \{D_j \,|\, D_j \in \times_{\underset{R_i \in R}{\cup}} \mathbb{D}_i, |D_j| > \mathcal{L}, D_{R_i} \in \mathbb{D}_{R_i}, |D_{R_i}| \le \mathcal{L}\} \end{aligned}$$

Thereby, the algorithm give that

$$\mathcal{L}^{new} := \min_{D_j \in \mathcal{D}} |D_j|,$$

where $\mathcal{L} < |D_j|$, $\mathcal{D} = \times_{\cup R_i \in R} \mathbb{D}_i$.

When $\Delta\mathcal{L} \neq 1$ for some l, the proof follows the same idea as above with the difference that there will be several different sizes of diagnoses included in for example T_l. ■

The next lemma states the result after the final call from Algorithm 1 to Algorithm 2 for a set of agents R.

Lemma 3. After computation of UpdateAgent$(R_n, R_{n-1}, \mathcal{L})$, *where* $R = \{R_1, \ldots, R_n\}$, *then if* $N^{R_n} \neq \emptyset$ *in agent* $R_n \in R$ *then*

$$N^{R_n} = \mathcal{D}^{mod,mc}. \tag{9}$$

◇

Proof: The initiation of

$$\mathcal{L}^{initiation} := \max_{A_i \in R} \min_{D_j \in \mathbb{D}_i} |D_j|$$

gives that $|\mathcal{D}_j^{mod}| \ge \mathcal{L}^{initiation}$, from this follows that the initiation of $\mathcal{L}$ does not result in the removal of any possible $\mathcal{D}_i^{mod}$ when using Algorithm 2.

If $N^{R_n} = \emptyset$ after call to UpdateAgent, then at the next computation of UpdateAgent$(R_n, \ldots)$, $N^{R_n} \neq \emptyset$ and $\mathcal{L} = \min_j\{|D_j| \,\big|\, D_j \in \mathcal{D}^{mod}\}$ where $\mathcal{D}^{mod} = \times_{\cup R_i \in R} \mathbb{D}_i$. From Lemma 2 is given that

$$N^{R_n} = \{D_j \,\big|\, \mathcal{L}^{l-1} < |D_j| \le \mathcal{L}^l, D_j \in \mathcal{D}^{mod}\} \tag{10}$$

therefore

$$N^{R_n} = \{D_j \,\big|\, |D_j| = \mathcal{L}, D_j \in \mathcal{D}\} = \mathcal{D}^{mod,mc}. \tag{11}$$

If $N^{R_n} \neq \emptyset$ then, if there has been no preceding R_k with $N^{R_k} \neq \emptyset$, then $\mathcal{L} = \mathcal{L}^{initiation}$ and (11) is true. Otherwise there is some latest preceding R_k where $N^{R_k} = \emptyset$ in the first traversal and

$$\mathcal{L} = \min_j\{|D_j| \,\big|\, D_j \in \mathcal{D}^{mod}\} \le |N_i^{R_n}|$$

which show (11) is true. ■

Finally the proof of Theorem 1.

Proof: From the algorithm follows directly that all $\mathcal{D}^{part,mc,k}$ are disjoint. Lemma 3 gives that $\mathcal{D}^{part,mc,k}$ are MGMCDS. ■

IV. Conclusions

There is an increasing number of systems that uses multiple agents to achieve some stated tasks. One such example is found in the vehicle industry where new vehicles might include several dozens of ECUs, which are used to control different parts of the vehicle. With the increasing complexity comes a higher demand for diagnosis, i.e. the system must be able to detect and localize faults in the whole system. Due to the increasing number of connections between the agents, this task puts new demands on the diagnostic system. One such demand is that the agents should be able to communicate with each-other to state diagnoses that are consistent with the knowledge stored in each agent.

It has been shown how an algorithm that uses the agents local diagnoses to state global diagnoses could be designed. To reduce the complexity, only the global diagnoses with minimal cardinality was considered. The algorithm uses the agents own processing power which reduced the need for a central diagnostic agent.

References

[1] R. Reiter, "A theory of diagnosis from first principles," *Artificial Intelligence*, vol. 32, no. 1, pp. 57–95, April 1987.

[2] J. Kurien, X. Koutsoukos, and F. Zhao, "Distributed diagnosis of networked, embedded systems," in *Thirteenth International Workshop on Principles of Diagnosis*, Semmering, Austria, May 2002.

[3] P. Fröhlich and W. Nejdl, "Resolving conflicts in distributed diagnosis," in *ECAI Workshop on Modelling Conflicts in AI*, Budapest, August 1996.

[4] N. Roos, A. ten Teije, and C. Witteveen, "A protocol for multi-agent diagnosis with spatially distributed knowledge," in *2nd Conference on Autonomous Agents and Mult-Agent Systems*, Australia, July 2003.

[5] J. de Kleer and B. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, no. 1, pp. 97–130, April 1987.

[6] W. Hamscher, L. Console, and J. de Kleer, Eds., *Readings in Model-Based Diagnosis*. Morgan Kaufmann Publishers, 1992.