Possible conflicts, ARRs, and conflicts

Belarmino Pulido Junquera and **Carlos Alonso González**
Departamento de Informática. Universidad de Valladolid
Edificio Tecnologías de la Información y las Telecomunicaciones
E-47011. Valladolid (Spain) ***e-mail*: {belar,calonso}@infor.uva.es**
Phone: +34 983 42 36 70 Fax: +34 983 42 36 71

Abstract. Consistency-based diagnosis is the most widely used approach to model-based diagnosis within the Artificial Intelligence community. It is usually carried out through an iterative cycle of behavior prediction, conflict detection, and candidate generation and refinement. Many approaches to consistency-based diagnosis have relied on some kind of on-line dependency-recording mechanism for conflict calculation. These techniques have had different problems, specially when applied to dynamic systems. Recently, off-line compilation of dependencies has been established as a suitable alternative approach. In this work we compare one compilation technique, based on the *possible conflict* concept, with results obtained with the classical on-line dependency recording engine as in GDE. Moreover, we compare possible conflicts with another compilation technique coming from the FDI community, which is based on analytical redundancy relations. Finally, we study the relationship between possible conflicts, analytical redundancy relations, and conflicts.

1 Introduction

For more than thirty years different techniques have been applied to diagnose systems in multiple domains. Diagnosis has been carried out through knowledge-based systems, case-based reasoning, model-based reasoning, and so on. This work is focused in the model-based approach to diagnosis. Moreover, we will only talk about diagnosis of physical devices [18].

More specifically, consistency-based diagnosis is the most widely used approach to model-based diagnosis within the Artificial Intelligence community (usually known as DX). It is a research field that has reported successful results in recent years [39, 7]. This approach has proven its maturity, both in theory, and in practice. On the one hand, the diagnosis process and the diagnosis results have been completely characterized from a logical point of view [32, 12], thus facilitating further comparison. On the other hand, consistency-based diagnosis has been successfully applied to a wide variety of domains such as automotive industry [3, 38], bio-medicine [20], nuclear plants [24], or ecology [37].

In such a framework, GDE [13] is the most well known implementation, and *de facto* paradigm. GDE organizes the diagnosis process as an iterative cycle of behavior prediction, conflict detection, and candidate generation and refinement. But conflict computation is a non-trivial step, which has deserved a lot of attention from the consistency-based diagnosis community. In GDE, the set of minimal conflicts is computed by means of an ATMS [11], which records on-line the set of correctness assumptions, or dependencies, used by the inference engine. It should be noticed that dependency-recording can be done forward (whenever new input data are introduced), or backward (when a discrepancy is found, such as in CAEN [2, 21], DYNAMIS [6], or TRANSCEND [25]). Another important feature of the GDE framework is that it calculates labels propagating values through constraints in every possible direction.

However, one problem related to on-line dependency-recording is that the set of labels needs to be computed each time a new different value is introduced. Another problem was found in the combined use of on-line dependency-recording together with qualitative models for diagnosing dynamic systems [17, 14]. Mainly for these reasons several research groups have looked for alternative methods to such a kind of on-line dependency-recording. On the one hand *state-based diagnosis* [36] has emerged as an alternative to *simulation-based diagnosis*, just for qualitative models. On the other hand, *topological methods* propose to explicitly use the structural description of the system to be diagnosed. This information is implicitly stated in the system description. Within this last approach, we make difference of two major trends: those methods that use other on-line dependency-recording than ATMS (by exploring causal-graphs [2, 24], signed directed graphs [26], or other topological and functional structures [5]), and those methods that perform off-line dependency-recording.

Last techniques are also known as compilation methods within the DX community. The main idea supporting this approach is that redundancy within the models can be found off-line. A similar idea was used in the Control Engineering community (or FDI), where Staroswiecki and Declerk proposed to use Analytical Redundancy Relations (ARRs for short), for fault detection and localization [34]. Given such a similarity, there is an ongoing interest from the DX and the FDI communities in comparing their approaches.

Between the FDI and AI proposals, Lunze and Schiller [23] were able to perform diagnosis using causal graphs associated with over-constrained systems. These systems were obtained from the logical formula in the models of the system.

Within the DX community we have found the following compilation techniques:

- Darwiche and Provan [10] characterized the set of diagnoses using the consequence concept [9], instead of using the conflict concept. Analyzing the system structure, those sub-systems which could lead to a diagnosis can be found off-line.
- Similar information is used by Steele and Leitch [35] to refine the set of candidates, in an adaptive approach to diagnosis [4].
- In DOGS, Loiez and Taillibert [22] proposed to localize, off-line, over-constrained sets of equations. They were looking for those sub-systems capable to become conflicts. The work done is con-

ceptually equivalent to that in [34], as it has been stated in [8].

- Fröhlich and Nejdl [15] used structural information two-fold: they analyzed the whole set of logical formula in the model to find sub-sets of formula capable to generate diagnosis, and they benefit from these sub-sets in order to refine the whole set of diagnosis candidates.
- Pulido and Alonso [27, 28] proposed to organize consistency-based diagnosis around the *possible conflict* concept. A possible conflict is a sub-system in system description which is capable to become a conflict, within the GDE framework.

In this work we revisit the compilation technique based on the *possible conflict* concept [27, 28]. Initially we summarize the characterization of that concept, in order to compare possible conflicts against real conflicts. Later on, we establish the relationship between possible conflicts and ARRs. Finally, we revisit the work by Cordier et al. [8] in order to compare conflicts and ARRs from a computational point of view.

Due to space limitations we do not compare possible conflicts and other compilation techniques from the DX community. Such a comparison can be found in [28, 30].

2 The possible conflict concept

Main assumptions in this work are that there is no structural fault, and it is possible to know beforehand the number and placement of available observations (sensors). An additional assumption is that the model of the system can be expressed as a set of constraints: quantitative or qualitative, linear or not, algebraic or not.

In Reiter's framework for model-based diagnosis [32] a minimal conflict identifies a set of constraints containing enough redundancy to perform diagnosis. In the most simple case, when constraints are made up of equations, a minimal conflict would denote a strictly over-determined system[1].

As it was mentioned in the previous Section, shared basis in compilation techniques is: the set of analytically redundant sub-systems, which can be used for diagnosis purposes, can be computed off-line. Moreover, it has been proven that GDE provides all the existing minimal conflicts. Since the set of possible conflicts tries to be a computational alternative to on-line dependency recording for conflict computation, we have imposed an additional requirement: over-constrained sub-systems should be the same as the set of minimal conflicts computed by GDE [2].

Finding analytical redundancy is a necessary but not a sufficient condition for a system to be suitable for consistency-based diagnosis purposes. The system must also be solved using local propagation alone[3]. To fulfill both requirements we have split the search process into two phases. First, we look for over-determined systems. Second, we check whether these systems can be solved using local propagation alone. To do so, we just need abstractions of model-description. For the sake of readability, below we include a summary of definitions the reader can find in [27, 28].

2.1 Searching for over-determined systems

We have represented the model in SD as a hyper-graph: $H_{SD} = \{V, R\}$ which is made up of:

- $V = \{v_1, v_2, \ldots, v_n\}$, the set of variables in the model. It is made up of observed OBS, and not observed or unknown variables, $NOBS$: $V = OBS \bigcup NOBS$.
- $R = \{r_1, r_2, \ldots, r_m\}$ is a family of sub-sets in V, where each r_k represents a constraint in the model, and it contains some model variables, observed and not observed ones.

We have called *Evaluation Chains* the over-constrained sub-systems in H_{SD} (in Appendix A the reader can find definitions for terminology in graphs and hyper-graphs c.f. [16, 1]):

Evaluation chain: $H_{ec} \subseteq H_{SD}$ is a partial sub-hypergraph in H_{SD}: $H_{ec} = \{V_{ec}, R_{ec}\}$, where $V_{ec} \subseteq V$, $R_{ec} \subseteq R$, and $X_{ec} = V_{ec} \cap NOBS$ is the set of unknowns in V_{ec}, and H_{ec} verifies:

1. H_{ec} is a connected hypergraph,
2. $V_{ec} \cap OBS \neq \emptyset$,
3. $\forall v_{no} \in X_{ec} \Rightarrow d_{H_{ec}}(v_{no}) \geq 2$,
4. let $G(H_{ec})$ be a bipartite graph made up of two kinds of nodes: $x \in X_{ec}$, and $r_{i_{ec}} \in R_{ec}$, such that two nodes are linked in $G(H_{ec})$ if and only if $x \in r_{i_{ec}}$. Then, $G(H_{ec})$ has a *matching* with maximal cardinality $m' = |X_{ec}|$ and $|R_{ec}| \geq m' + 1$.

Figure 1 shows a classical example in consistency-based diagnosis. In order to make difference of components and constraints, we will use capital letters for components, and small letters for constraints in their models. m_i and a_j denote the models of multipliers and adders, respectively. Each model is made up of just one constraint; for instance, $m_1 = \{A, C, X\}$. Whenever a model has more than one constraint, indices are used to distinguish them. The related hyper-graph is

$$H_{polybox} = \{\{A, B, C, D, E, F, G, X, Y, Z\}, \{m_1, m_2, m_3, a_1, a_2\}\}$$

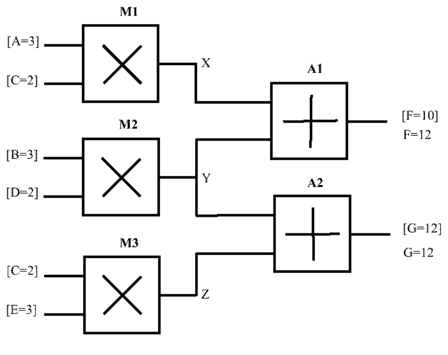

Figure 1. Classical polybox example in the consistency-based diagnosis. Observed values are in brackets. $\{X, Y, Z\}$ are non-observed values.

Since we are interested in minimal conflicts, only minimal evaluation chains, MEC for short, are useful.

Minimal Evaluation Chain : H_{ec} is a minimal evaluation chain if there is no evaluation chain $H'_{ec} \subset H_{ec}$.

The set of minimal Evaluation chains, SMEC, is built based on the algorithms: *build-every-mec()*, *build-mec()*, and *justify()* which perform depth-first search in H_{SD} using backtracking. All these algorithms can be found in Appendix B. In the polybox example, these

[1] In an over-determined system the number of equations, e, is greater than the number of unknowns, u: $e \geq u + 1$. In a strictly over-determined system, $e = u + 1$.

[2] For this reason, we always assume that we have the same model (system description or *SD* in Reiter's terminology) as GDE has.

[3] Current consistency-based diagnosis systems do not impose that constraint [19]. In [30] we extended the possible conflict concept to deal with such (cyclical) configurations.

algorithms have found three MECs:

$$\begin{array}{l} H_{ec_1} = \{\{A, B, C, D, F, X, Y\}, \{m_1, m_2, a_1\}\} \\ H_{ec_2} = \{\{B, C, D, E, G, Y, Z\}, \{m_2, m_3, a_2\}\} \\ H_{ec_3} = \{\{A, C, E, F, G, X, Y, Z\}, \{m_1, a_1, a_2, m_3\}\} \end{array}$$

2.2 Can an evaluation chain be solved?

A minimal conflict is a strictly over-determined system that we want to solve using local propagation alone. However, the hyper-graph has not enough information about how each constraint can be solved. To tackle this problem, we create an AND-OR graph for each minimal evaluation chain. In such a graph, there is one or more AND-OR arcs for each hyper-arc in the MEC. Each AND-OR arc represents one way the hyper-arc could be solved. In fact, to solve a MEC, we should select one AND-OR arc from each constraint. As a consequence, choosing different AND-OR arcs from the AND-OR graph generates different ways of solving the MEC. Moreover, the over-determined system can only be solved using local propagation criteria. Each one of the different ways of solving a MEC is called a Minimal Evaluation Model, or MEM.

For instance, each constraint (m_i or a_i) used to model the polybox system provides three different interpretations to the AND-OR graph:

$$m_i(v_{out}, v_{in_1}, v_{in_2}) \Rightarrow \left\{ \begin{array}{l} m_{i_1} \equiv v_{out} = v_{in_1} \times v_{in_2} \\ m_{i_2} \equiv v_{in_1} = v_{out}/v_{in_2},\ if\ v_{in_2} \neq 0 \\ m_{i_3} \equiv v_{in_2} = v_{out}/v_{in_1},\ if\ v_{in_1} \neq 0 \end{array} \right.$$

Interpretations for a constraint are usually obtained when applying the invertibility criterion. Nevertheless, there are additional criteria. Appendix D shows constraints used to model a physical system made up of tanks, pumps and valves. Constraints $tr1_3, t2_3, tr2_5$ are used to compute the mass in a tank. In such kind of constraint, just one interpretation is allowed, since we have taken an integration approach:

$m_T(t) = \int m'_T(t-1)dt + m_T(t-1)$

This interpretation can not be reversed. Hence, additional concepts are necessary to define a Minimal Evaluation Model.

Given the relation between $r_{i_{ec}} \in R_{ec}$, and the set of AND-OR arcs $r_{i_{k_{em}}}$, derived from $r_{i_{ec}}$, we can state the following proposition.

Proposition 1 *Let $AOG(H_{ec}) = \{V_{em}, R_{em}\}$ be the AND-OR graph obtained from $H_{ec} = \{V_{ec}, R_{ec}\}$ applying the local resolution criterion, where:*

- $V_{em} = V_{ec}$,
- $\forall r_{i_{ec}} \in R_{ec} \Rightarrow \exists r_{i_{k_{em}}} \in R_{em}, k \geq 1$

Then, $r_{i_{ec}} \in R_{ec}$ induces a partition in R_{em}.

Proof: Each $r_{i_{ec}} \in R_{ec}$ induces an equivalence class in R_{em}. By definition, it induces a partition too.

Leaf node: v_i is a leaf node in graph H iff $\hat{\Gamma}_{v_i}^{-1} = 0$.

Discrepancy node: v_i is a discrepancy node in graph H iff

- $(d_H^-(v_i) \geq 2\ \wedge\ v_i \in NOBS)$, or
- $(d_H^-(v_i) \geq 1\ \wedge\ v_i \in OBS)$

That is, a leaf node has no predecessors, and a discrepancy node can be found in two different ways: estimating an observed variable, or doing a double estimation for an unknown variable.

Minimal Evaluation Model: A partial AND-OR graph, $H_{mem} \subseteq AOG(H_{ec})$, where $H_{mem} = \{V_{mem}, R_{mem}\}$, is a minimal Evaluation model iff:

1. R_{mem} is a minimal hitting-set for the partition induced by $r_{i_{ec}} \in R_{ec}$ in R_{em},
2. $(\forall v_i \mid v_i \in V_{mem}$ and v_i is a leaf node$) \Rightarrow v_i \in OBS$,
3. $\exists_1 x_j \in V_{mem} \mid x_j$ is a discrepancy node,
4. if x_j is a discrepancy node, then there exists a directed and acyclic path in $H_{mem} : \{x_i, x_{i+1}, \ldots, x_{i+k}, x_j\}$ from each node x_i to x_j.

Algorithms used to calculate every MEM for each MEC: *build-every-mem()*, and *build-mem()*, are given in Appendix C. These algorithms are exhaustive too, since they perform depth-first search using backtracking. For instance, MEC H_{ec_1} has a related AND-OR graph:

$$\begin{array}{l} AOG(H_{ec_1}) = \{\{A, B, C, D, F, X, Y\}, \\ \{m_{1_1}, m_{1_2}, m_{1_3}, m_{2_1}, m_{2_2}, m_{2_3}, a_{1_1}, a_{1_2}, a_{1_3}\}\} \end{array}$$

Given H_{ec_1} and the set of available interpretations in $AOG(H_{ec_1})$, algorithm *build-mem()* is able to find seven different MEMs[4]:

MEMs	**Equivalent to evaluate the expression**
$\{m_{1_1}, m_{2_1}, a_{1_1}\}$	$F_{obs} \equiv F_{pred} = A \times C + B \times D$
$\{m_{1_1}, m_{2_1}, a_{1_2}\}$	$X_{pred_1} = A \times C \equiv X_{pred_2} = F - B \times D$
$\{m_{1_2}, m_{2_1}, a_{1_2}\}$	$A_{obs} \equiv A_{pred} = (F - B \times D)/C,\ if\ C \neq 0$
$\{m_{1_3}, m_{2_1}, a_{1_2}\}$	$C_{obs} \equiv C_{pred} = (F - B \times D)/A,\ if\ A \neq 0$
$\{m_{1_1}, m_{2_1}, a_{1_3}\}$	$Y_{pred_1} = F - (A \times C) \equiv Y_{pred_2} = B \times D$
$\{m_{1_1}, m_{2_2}, a_{1_3}\}$	$B_{obs} \equiv B_{pred} = (F - A \times C)/D,\ if\ D \neq 0$
$\{m_{1_1}, m_{2_3}, a_{1_3}\}$	$D_{obs} \equiv D_{pred} = (F - A \times C)/B,\ if\ B \neq 0$

It should be noticed that a MEC would provide no MEM if the over-determined system can not be solved using available interpretations and local propagation. In [31] the reader can find additional information on how temporal information has been included in this framework and one example of a MEC which can not provide any MEM.

Once summarized the possible conflict concept, next section studies the relationship between MECs, and MEMs, which are computed off-line, and real conflicts computed on-line.

3 Conflicts and possible conflicts

If evaluated, a MEM could lead to discrepancy, i.e., it could lead to a conflict. However, the set of MEM is computed off-line, without any model evaluation. And conflicts would appear only when observations are introduced and the evaluation model is computed. So, we have introduced the following concept:

Possible conflict: The set of constraints in a Minimal Evaluation Chain giving rise to, at least, one Minimal Evaluation Model.

For example, in the polybox system in Figure 1, there are three possible conflicts: $\{\{m_1, m_2, a_1\}, \{m_1, a_1,\ a_2, m_3\}, \{m_2, m_3, a_2\}\}$, because every MEC has, at least, one MEM.

In such a case, where component models are made up of only one relation, the set of possible conflicts is equivalent to the set of minimal conflicts in Reiter's terminology computed on-line by GDE, whatever the faults and whatever the set of available observations.

At this point it is necessary to answer the following question: is the set of possible conflicts equivalent to the set of minimal conflicts computed on-line by GDE? In order to answer, we need additional definitions:

$P(S)$**:** is the set of subsets in S;

[4] Since the MEM will have the same set of variables as MEC, we just include the set of interpretations.

$model : COMPS \rightarrow P(R_{SD})$: $model(C)$ identifies the family of relations modelling C behavior;

$comp : R_{SD} \rightarrow COMPS$: $r_i \rightarrow comp(r_i) = \{C \mid r_i \in model(C)\}$:
$comp(r_i)$ indicates the component containing relation r_i in its model.

Proposition 2 *Let co be a minimal conflict found by GDE, and co is related to a discrepancy in $v \in V_{SD}$: there is a minimal evaluation chain, $H_{ec} = \{V_{ec}, R_{ec}\}$, such that:*
$v \in V_{ec}$ *and* $co = \bigcup_{r_i \in R_{ec}} comp(r_i)$

Proof: GDE solves a minimal over-determined system to find a minimal conflict related to v [19]. Since *build-every-mec()* performs exhaustive search, it is able to find every minimal over-determined system in H_{SD}. Hence, it will find that over-determined system too.

Hence, once GDE finds a minimal conflict, *build-every-mec()* will find a MEC containing the same set of constraints which were used to find a conflict. Those constraints belong to the same set of components.

Proposition 3 *Let co be a minimal conflict found by GDE, and co is related to a discrepancy in $v \in V_{SD}$: there is a minimal evaluation model, $H_{me} = \{V_{em}, R_{em}\}$, that can obtain a discrepancy in v, and* $v \in V_{em}$ *and* $co = \bigcup_{r_i \in R_{em}} comp(r_i)$

Proof: By proposition 2, there is a MEC related to co, such that:

$$co = \bigcup_{r_i \in R_{ec}} comp(r_i)$$

Moreover *build-every-mem()* performs an exhaustive search too. Therefore, it will find every MEM related to such MEC, i.e., every possible way the MEC can be solved. Hence, it will find the over-determined system used to obtain the minimal conflict. Also, each $r_{i_k} \in R_{em}$ is an interpretation for some $r_i \in R_{ec}$. Hence:

$$co = \bigcup_{r_{i_k} \in R_{em}} comp(r_i)$$

At least one of the MEM related to the CEM will find a discrepancy in v, in the same way the GDE does.

Unfortunately, the number of MEMs for each MEC is exponential in the average number of interpretations for each hyper-arc in the MEC. Due to practical reasons we just select one MEM related to a MEC. Based on that MEM, we build an executable model which is used for fault detection. In [31] the reader can find a detailed description of how possible conflicts can be used to perform consistency-based diagnosis for both static and dynamic systems.

Nevertheless, it is still possible to claim that the set of possible conflicts is theoretically equivalent to the set of conflicts found on-line by means of GDE. We will show this fact in next two propositions.

Proposition 4 *If H_{ec} is a MEC, H_{em} is one of its MEMs and the evaluation of the executable model associated to H_{em} generates a discrepancy in $v \in V_{em}$, then GDE will find a discrepancy in v.*

Proof: There is a discrepancy in v related to the evaluation of a MEM. The MEM is an strictly over-determined system. Moreover, GDE finds any discrepancy related to any minimal over-determined system. Hence, it will find the discrepancy in v too.

This proposition always holds. Unfortunately, the converse does not hold universally, because we can not guarantee for an arbitrary set of non-linear constraints that every MEM for a MEC will provide the same solution for a given set of observations [40]. This assumption should be stated in the following way:

Equivalence assumption : Every MEM in a MEC provides the same set of solutions for any given set of input observations.

Now, it is possible to define the following proposition:

Proposition 5 *If GDE finds a minimal conflict, co, related to a discrepancy in v, and* ***the equivalence assumption holds*** *for a H_{ec} containing v, then the possible conflict related to H_{ec} will be confirmed as a minimal conflict.*

Proof: The proof is straightforward based on propositions 2, and 3.

4 Comparing possible conflicts, conflicts, and ARRs

As previously mentioned, there is an on-going research interest from the DX and FDI communities in comparing their approaches. Recently, Cordier et al. [8] proposed a common framework to compare conflicts and ARRs [34, 33]. In that trend, we compare ARRs and possible conflicts considering the way they are computed. Afterwards, we discuss results in [8] and extract some conclusions.

4.1 Possible conflicts and ARRs

The set of ARRs is obtained from the unique canonical decomposition of the structural description of the system into under-determined, just-determined, and over-determined sets of constraints. The canonical decomposition is based on finding a complete matching, w.r.t. unknown variables, in the bipartite graph associated to the structural description of the system. Combination of just-determined systems together with redundant relations is the basis for an *Analytical Redundancy Relation*[34].

Each complete matching can be considered as a causality assignment, but it is necessary to obtain a causal matching for the over-determined system, from the set of causal matchings satisfying the invertibility condition [33]. Each ARR can be solved and used for diagnosis purposes once observed values are introduced.

It should be noticed that all the steps, except the solving one, could be done off-line. Hence, computing ARRs is a compilation technique in FDI. And, it seems obvious that strong similarities do exist between the way ARRs and possible conflicts are computed.

- Both methods search for over-determined sub-systems. Direct or deduced ARRs can be used to estimate a value for an observed variable in the system. Moreover, algorithms used for computing MEC, can be used to obtain the whole set of over-determined sub-systems[5]. Hence, the algorithms will find an evaluation chain with the same set of constraints as of the ARR.
- An ARR need a causal matching, because not every causality assignment can be done in the complete matching. In the same way, AND-OR arcs are introduced to limit the ways an hyper-arc can be solved. It seems obvious that one of the evaluation models for an evaluation chain will be equivalent to the causal matching in the ARR.

[5] It is straightforward to modify algorithm *Justify()* to search for any over-determined system.

- The set of evaluation models for an evaluation chain are built based on local propagation criterion, i.e., the evaluation model does not contain any cycle. This condition has been imposed in the ARR approach too. For this reason, the ARR is obtained once graph reduction, by means of loop elimination, has been done in the causal graph [33]. This step is equivalent to loop elimination in the possible conflict approach [29].

However, there are some differences:

- Staroswiecki et al. [33] assume that in an over-determined system the set of unknowns can be computed in different ways, using constraints and known values, and "deduced redundancy relations are obtained writing that all these results have to be the same". This assumption is the same as the equivalence assumption in the previous section.
  As mentioned above, that assumption is never done in GDE while computing minimal conflicts, because the assumption does not hold universally for physical systems made up of general non-linear constraints [40]. Therefore, based on propositions 4 and 5, it can not be claimed that model-based diagnosis relying upon ARRs and consistency-based diagnosis using conflicts will provide always the same set of results. Results obtained using ARRs would be the same as of those obtained using just one MEM for each MEC. These results can be sub-optimal, w.r.t. the number of detected conflicts, unless the equivalence assumption holds.
- Moreover, *build-every-mec()* provides the whole set of minimal evaluation chains, because we look for minimal conflicts. This is not guaranteed in the original ARR approach, which should be revised to find just minimal ARRs.

4.2 Discussion

Cordier et al. [8] defined the *support* for an ARR as "the set of components involved in the ARR". This term was also called "potential R-conflict", because of their Proposition 4.1:

> "Let OBS be a set of observations for a system modeled by SM (resp. SD). There is an identity between the set of minimal R-conflicts for OBS and the set of minimal potential R-conflicts associated to the ARRs which are not satisfied by OBS."

As stated in the previous section, we think it is necessary to make three explicit assumptions to guarantee that such a conclusion holds universally:

- the equivalence assumption holds,
- the set of ARRs is built based on minimality criteria, and
- we have a component-oriented behavior description of the system, but minimality is considered w.r.t. sets of constraints.

Regarding first two conditions, it seems obvious that proposition 5 in Section 3 is equivalent to proposition 4.1. in [8] when both assumptions hold. Third assumption must be taken into account when behavioral models are made up of more than one constraint. Minimality w.r.t. sets of constraints is needed because not every possible conflict is equivalent to a minimal conflict in Reiter's framework. We will illustrate this using the system in Figure 2. The system is made up of common components in process industry such as tanks, pumps, valves, and so on.

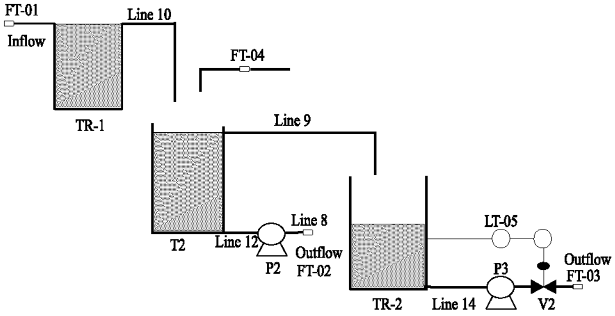

Figure 2. Scheme of the system to be diagnosed. Measured variables are flows $FT01 = f_1^*$, $FT02 = f_8^*$, $FT03 = f_7^*$, and $FT04 = f_{11}^*$; level of tank $LT05 = h_{TR2}^*$, and the value of the control action on valve $V_2 = u_2$ at the output of tank $TR2$.

Its related hyper-graph can be described as:

$$
\begin{aligned}
&\mathbf{H}_{SD} = \{V_{SD}, R_{SD}\};\\
&\mathbf{V}_{SD} = \{OBS \cup NOBS\};\\
&\mathbf{O}BS = \{f_1^*, f_7^*, f_8^*, f_{11}^*, h_{TR_2}^*\};\\
&\mathbf{N}OBS = \{f_9, f_{10}, f_{12}, f_{14}, m'_{TR_1}, m_{TR_1}, h_{TR_1}, m'_{T_2}, m_{T_2},\\
&h_{T_2}, m'_{TR_2}, m_{TR_2}, h_{TR_2}, \Delta P_{P_2}, \Delta P_{P_3}, P1_{TR_1}, P2_{TR_1}, P1_{T_2},\\
&P2_{T_2}, P1_{TR_2}, P2_{TR_2}, u_{cont_2}\}\\
&\mathbf{R}_{SD} = \{tr1_1, tr1_2, tr1_3, tr1_4, t2_1, t2_2, t2_3, t2_4, t2_5, p2_1, p2_2,\\
&p2_3, p3_1, p3_2, p3_3, v2_1, v2_2, tr2_1, tr2_2, tr2_3, tr2_4, tr2_5, tr2_6\}
\end{aligned}
$$

The meaning for each equation above can be found in Appendix D. We have used common equations for computing mass balances, overflows, and so on. Analyzing the system we have found three possible conflicts. The reader should notice that PC_3 is minimal w.r.t. constraints, but not minimal w.r.t. components.

PC$_i$	**Components**
$\{tr1_1, tr1_2, tr1_3, tr1_4, t2_1, t2_2, t2_3, t2_4, t2_5, p2_1, p2_2, p2_3\}$	$\{TR_1, T_2, P_2\}$
$\{tr2_1, tr2_3, p3_1, p3_2, p3_3, v2_1, v2_2\}$	$\{TR_2, P_3, V_2\}$
$\{tr1_1, tr1_2, tr1_3, tr1_4, t2_1, t2_2, t2_3, t2_5, p2_3, tr2_4, tr2_5, tr2_6, p3_3, v2_1\}$	$\{TR_1, T_2, P_2, TR_2, P_3, V_2\}$

5 Conclusions

In this paper we have shown that compilation of dependencies by means of the possible conflict approach is theoretically equivalent to on-line dependency recording in GDE. However, it is not possible to claim that, in practice, consistency-based diagnosis using possible conflicts provides the same results as GDE does, unless the equivalence assumption holds.

We have found out that the model of an ARR is equivalent to some evaluation model for an evaluation chain. Since we select just one MEM for each MEC for practical reasons, we conclude that both approaches can obtain equivalent results (assuming ARRs are computed based on minimality criteria).

Finally, we have concluded that Proposition 4.1 in [8] need to be revised taking into account results in propositions 4 and 5, and considering minimality criteria w.r.t. constraints.

Acknowledgements

Authors wants to thank Louise Travé-Massuyès for critical reading of the original manuscript, and three anonymous referees for their valuable comments.

REFERENCES

[1] C. Berge, *Hypergraphs. Combinatorics of Finite Sets*, North-Holland, Amsterdam, 1989.

[2] K. Bousson, J.-P. Steyer, L. Travé-Massuyès, and B. Dahhou, 'From a heuristic-based to a model-based approach for monitoring and diagnosis of biological processes', *Engineering Applications of Artificial Intelligence*, **11**, 447–493, (1998).

[3] F. Cascio, L. Console, M. Guagliumi, M. Osella, A. Panati, S. Sottano, and D. Theseider Dupre, 'Generating on-board diagnostics of dynamic automotive systems based on qualitative models', in *Proceedings of the Tenth International Workshop on Principles of Diagnosis (DX-99)*, pp. 27–35, Loch Awe, Scotland (UK), (1999).

[4] M.J. Chantler, G.M. Coghill, Q. Shen, and R.R. Leitch, 'Selecting tools and techniques for model-based diagnosis', *Artificial Intelligence in Engineering*, **12**, 81–98, (1998).

[5] L. Chittaro, G. Guida, C. Tasso, and E. Toppano, 'Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems: A case study in diagnosis', *IEEE Transactions on Systems, Man and Cybernetics*, **23**, 1718–1751, (1993).

[6] L. Chittaro, R. Ranon, and J. Lopez Cortes, 'Ship over troubled waters: Functional reasoning with influences applied to the diagnosis of a marine technical system', in *Proceedings of the Seventh International Workshop on Principles of Diagnosis (DX-96)*, pp. 69–78, Val Morin, Quebec, Canada, (1996).

[7] L. Console and O. Dressler, 'Model-based diagnosis in the real world: lessons learned and challenges remaining', in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pp. 1393–1400, Stockholm, Sweeden, (1999).

[8] M.O. Cordier, P. Dague, M. Dumas, F. Levy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès, 'A comparative analysis of ai and control theory approaches to model-based diagnosis', in *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI 2000)*, pp. 136–140, Berlin, Germany, (2000).

[9] A. Darwiche, 'Model-based diagnosis using structured system descriptions', Technical report 97-07, Department of Mathematics, American University of Beirut, Beirut, Lebanon, (1997).

[10] A. Darwiche and G. Provan, 'Exploiting system structure in model-based diagnosis of discrete-event systems', in *Proceedings of the Seventh International Workshop on Principles of Diagnosis (DX-96)*, pp. 93–105, Val Morin, Quebec, Canada, (1996).

[11] J. de Kleer, 'Problem solving with the ATMS', *Artificial Intelligence*, **28**, 197–224, (1986).

[12] J. de Kleer, A.K. Mackworth, and R. Reiter, 'Characterizing diagnosis and systems', in *Readings in Model Based Diagnosis*, 54–65, Morgan-Kauffman Pub., San Mateo, (1992).

[13] J. de Kleer and B.C. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, **32**, 97–130, (1987).

[14] O. Dressler and P. Struss, 'The consistency-based approach to automated diagnosis of devices', in *Principles of Knowledge Representation*, ed., Gerhard Brewka, 269–314, CSLI Publications, Standford, (1996).

[15] P. Froelich and W. Nejdl, 'A static model-based engine for model-based reasoning', in *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence (IJCAI-97)*, pp. 446–471, Nagoya, Japan, (1997).

[16] M. Gondran and M. Minoux, *Graphs and Algorithms*, John Wiley and Sons, New York, 1984.

[17] T. Guckenbiehl and G Schaefer-Richter, 'SIDIA: Extending prediction based diagnosis to dynamic models', in *Proceedings of the First International Workshop on Principles of Diagnosis (DX-90)*, pp. 74–82, Stanford, California, USA, (1990).

[18] W.C. Hamscher, L. Console, and J. de Kleer (Eds.), *Readings in Model based Diagnosis*, Morgan-Kaufmann Pub., San Mateo, 1992.

[19] G. Katsillis and M.J. Chantler, 'Can dependency-based diagnosis cope with simultaneous equations?', in *Proceedings of the Eigth International Workshop on Principles of Diagnosis (DX-97)*, pp. 51–59, Le Mont Saint Michel, France, (1997).

[20] M. Lafon, J. Pastor, L. Travé-Massuyès, B. Doyon, J.F. Demonet, and P. Celsis, 'Qualitative modeling of cerebral information propagation mechanisms', in *Proceedings of the Third International Conference on Computational and Neurosciences*, volume 2, pp. 21–23, Research Triangle Park, USA, (1998).

[21] A. Ligeza and B. Gorny, 'Systematic conflict generation in model-based diagnosis', in *Proceedings of the*, pp. 1103–1108, (2000).

[22] E. Loiez and P. Taillibert, 'Polynomial temporal band sequences for analog diagnosis', in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pp. 474–479, Nagoya, Japan, (1997).

[23] J. Lunze and F. Schiller, 'Logic-based process diagnosis utilising the causal structure of dynamical systems', in *Proceedings of the Artificial Intelligence in Real-Time Control (IFAC/IFIP/IMACS)*, pp. 649–654, Delft, Holland, (1992).

[24] P.J. Mosterman, *Hybrid dynamic systems: a hybrid bond graph modeling paradigm and its applications in diagnosis*, Phd thesis, Vanderbilt University, Nashville, Tennessee, USA, May 1997.

[25] P.J. Mosterman and G. Biswas, 'Monitoring, prediction and fault isolation in dynamic physical systems', in *Proceedings of the Fourteenth AAAI National Conference on arificial Intelligence (AAAI-97)*, pp. 100–105, Providence, Rhode Island, USA, (1997).

[26] O.O. Oyeleye, *Qualitative modeling of continuous chemical processes and applications to fault diagnosis*, Phd thesis, Department of Chemical Engineering, M.I.T., Cambridge, Massachusetts, USA, 1990.

[27] B. Pulido and C. Alonso, 'Possible conflicts instead of conflicts to diagnose continuous dynamic systems', in *Proceedings of the Tenth International Workshop on Principles of Diagnosis (DX-99)*, pp. 234–241, Loch Awe, Scotland (UK), (1999).

[28] B. Pulido and C. Alonso, 'An alternative approach to dependency-recording engines in consistency-based diagnosis', in *Artificial Intelligence: Methodology, Systems, and Applications. Ninth International Conference (AIMSA-00)*, volume 1904 of *Lecture Notes in Artificial Intelligence*, 111–120, Springer Verlag, Berlin, Germany, (2000).

[29] B. Pulido and C. Alonso, 'Dealing with cyclical configurations in mordred', in *IX Conferencia Nacional de la Asociación Española de Inteligencia Artificial, (CAEPIA-01)*, pp. 983–992, Gijon, Spain, (2001).

[30] B. Pulido and C. Alonso, 'Revisión del concepto de posible conflicto como técnica de pre-compilación', *Revista Iberoamericana de Inteligencia Artificial*, 41–53, (2001).

[31] B. Pulido, C. Alonso, and F. Acebes, 'Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models', in *Engineering of Intelligent Systems. Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2001)*, volume 2070. of *Lecture Notes in Artificial Intelligence*, pp. 135–144, Budapest, Hungary, (2001).

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

[33] M. Staroswiecki, S. Attouche, and M.L. Assas, 'A graphic approach for reconfigurability analysis', in *Proceedings of the Tenth International Workshop on Principles of Diagnosis (DX-99)*, pp. 250–256, Loch Awe, Scotland (UK), (1999).

[34] M. Staroswiecki and P. Declerk, 'Analytical redundancy in non linear interconnected systems by means of structural analysis', in *Proceedings of the IFAC Advanced Information Processing in Automatic Control (AIPAC-89)*, pp. 51–55, Nancy, France, (1989).

[35] A.D. Steele and R.R. Leitch, 'A strategy for time-constraint qualitative parameter identification', in *Proceedings of the Computational Engineering in Systems Applications (CESA IMACS-IEE/SMC Multiconferece)*, Lille, France, (1996).

[36] P. Struss, 'Fundamentals of model-based diagnosis of dynamic systems', in *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence (IJCAI-97)*, pp. 480–485, Nagoya, Japan, (1997).

[37] P. Struss and U. Heller, 'Model-based support for water treatment', in *Qualitative and Model Based Reasoning for Complex Systems and their Control, Workshop KRR-4 at the Sixteenth International Joint Conference on Artifical Intelligence (IJCAI-99)*, pp. 84–90, Stockholm, Sweeden, (1999).

[38] P. Struss, M. Sachenbacher, and C.M. Carlén, 'Insights from building a prototype for model-based on-board diagnosis of automotive systems', in *Proceddings of the Eleventh International Workshop on Principles of Diagnosis (DX-00)*, pp. 201–212, Morelia, Mexico, (2000).

[39] L. Travé-Massuyès and R. Milne, 'Gas-turbine condition monitoring using qualitative model-based diagnosis', *IEEE Expert*, **12**, 22–31, (1997).

[40] B.C. Williams and B. Millar, 'Automated decomposition of model-based learning problems', in *Proceedings of the Seventh International Workshop on Principles of Diagnosis (DX-96)*, pp. 258–266, Val Morin,

Quebec, Canada, (1996).

A Graph and hyper-graph notation

$H = [V, E]$	Hyper-graph H, made up V: nodes, and E: a family of sub-sets in V
Γ_i	Successors for node i
Γ_i^{-1}	Predecessors for node i
$d_H(i)$	Degree for node i in H
$d_H^+(i),\ d_H^-(i)$	Output and input demi-degree for node i in H

Bipartite graph: $G = [V, E]$ is a bipartite graph if there are two disjoints parts in $V = S \cup T$, and edges in E are always directed from S to T.

Matching: A matching M in $G = [V, E]$ is a subset of E such that no two arcs in M share a common vertex incident to them.

B Algorithms for computing the set of minimal evaluation chains

```
Algorithm build-every-mec (SMEC) is
  SMEC: set of MEC; { Each MEC is a set of constraints}
  available, to-be-justified, justified, chain: set of constraints;
  R, R2: constraint;
Begin
 available := Constraints-in($H_{SD}$);
 while available $\neq \emptyset$ do
  R := Select-constraint(available);
  chain := $\emptyset$;
  available := available \ {R};
  build-mec (SMEC, chain, R, available);
 end while
End
```

```
Algorithm build-mec (SCEM, chain, R, available) is
Begin
 Insert R in chain;
 to-be-justified := R.nobs;
 justified := $\emptyset$;
 Justify (SMEC, chain, to-be-justified, justified, available);
End
```

```
Algorithm Justify (SMEC, chain, to-be-justified, justified, available) is
  v: unknown variable;
  related: set of constraints;
Begin
 if to-be-justified = $\emptyset$ then
  if there is no subset of chain in SMEC then
   Erase chain supersets from SMEC;
   Insert chain in SMEC;
  end if { Only minimal chains are included in SMEC.}
 else
  v := select-variable (to-be-justified);
  related := R | R $\in$ available and v $\in$ R.nobs;
  while related $\neq \emptyset$ do
   R1 := select-r (related);
   related := related \ {R1};
   chain2 := chain $\cup$ {$R1$};
   Justified2 := Justified $\cup${$v$};
   to-be-justified2 := (to-be-justified \ v) $\cup$ (R1.nobs \ justified2 };
   available2 := available \ R1;
   Justify (SMEC, chain2, to-be-justified2, justified2, available2);
  end while
 end if
End
```

C Algorithms for computing the set of minimal evaluation models

```
Algorithm build-every-mem (SMEC, SMEM) is
 Begin
  for chain = each MEC in SMEC do
   for R = each constraint in chain do
    for I = each interpretation for R do
     model := {$I$};
     to-be-justified:= I.nobs;
     justified := $\emptyset$;
     chain := chain \ {$R$};
     build-mem (model, chain, to-be-justified, justified, SMEM);
    end for
   end for
  end for
 End
```

```
Algorithm build-mem (model, available, to-be-justified, justified, SMEM) is
   Begin
 if to-be-justified = $\emptyset$ and available = $\emptyset$ and $\exists_1$ discrepancy node in model then
  Insert model in SMEM;
 end if
 else
  for S = each constraint in available do
   if S.nobs $\cap$ to-be-justified = $\emptyset$ then
    for I2 = each interpretation for S do
     if head(I2) $\cap$ to-be-justified $\neq \emptyset$ then
      Insert {$I2$} in model;
      available := available \ {$S$};
      to-be-justified := (to-be-justified \ head(I2)) $\cup$ tail(I2).nobs;
      Insert head(I2) in justified;
      Build-mem (model, available, to-be-justified, justified, SMEM);
     end if
    end for
   end if
  end for
 end if
End
```

D Constraints used to model the hydraulic system

Constraints	Represent
$tr1_1, t2_1, tr2_4$	Mass balance in T: $m'_T = \sum f_{in} - \sum f_{out}$
$tr1_2, t2_2$	Overflow in T: $f_{out} = \sqrt{k \cdot (h_T - h_{ext})}$
$tr1_3, t2_3, tr2_5$	Mass: $m_T(t) = \int m'_T(t-1)dt + m_T(t-1)$
$tr1_4, t2_5, tr2_6$	Height in T: : $h_T = k_1 \cdot \frac{m_T}{A_T}$
$t2_4, tr2_2$	Pressure at bottom: $P_{T_1} = k_2 \cdot h_T + P_{atm}$
$p2_1, p3_2$	Pump load curve in P: $\Delta P_P = tablePQ(f_{out})$
$p2_2, p3_1$	Outflow in T: $f_{out} = \sqrt{k_3 \cdot \frac{(P_{T_1} + \Delta P_P - P_2)}{k_4}}$
$p2_3, p3_3, v2_1$	Flow out of tank: $f_{in} = f_{out}$
$tr2_1$	Control: $u = PID(h_T)$
$v2_2$	Flow through a valve: $f_{out} = \sqrt{k_5 \cdot \frac{(P_{T_2} - P_{atm})}{k_6 + (\frac{100}{u})^2}}$