arXiv:1411.4616v1 [cs.AI] 17 Nov 2014

A Note on Systematic Conflict Generation in CA-EN-type Causal Structures

Antoni Ligęza[*]

LAAS du CNRS, 7, Av. du Colonel Roche, 31077 Toulouse Cedex, France

E-mail: `ligeza@laas.laas.fr`,

Abstract

This paper is aimed at providing a very first, more "global", systematic point of view with respect to possible conflict generation in CA-EN-like causal structures. For simplicity, only the outermost level of graphs is taken into account. Localization of the "conflict area ", diagnostic preferences, and bases for systematic conflict generation are considered. A notion of *Potential Conflict Structure* (*PCS*) constituting a basic tool for identification of possible conflicts is proposed and its use is discussed.

1 Introduction

Diagnostic reasoning is an activity oriented towards detection of faulty behaviour and its explanation, i.e. isolation of faulty components responsible for the observed misbehaviour of the analyzed system. Model-based diagnosis is based on explicit system model applied for diagnostic inference. A widely accepted approach consists in *consistency-based* reasoning where the analysis is aimed at regaining consistency of the predicted model output with current observations by retracting some of the assumptions about correct behaviour of certain components. The sets of elements suspected to contain at least one faulty component are identified by detecting inconsistency between the observed and predicted behaviour. Such sets, called *conflict sets* are basic products for generating diagnoses.

A complete diagnostic procedure following the consistency-based approach should cover:

- detection and localization of misbehaviour,
- restriction of the search area (hierarchical fault diagnosis),
- systematic conflict generation, taking into account that:
  - *all* conflicts should be found, but
  - only *minimal* conflicts should be generated,

[*] On leave from: Institute of Automatics AGH, al. Mickiewicza 30, 30-059 Kraków, Poland; e-mail: `ali@ia.agh.edu.pl`. The author's stay in LAAS du CNRS was supported by a MENESRIP-DAEIF scholarship No.: 174755 K through CIES.

- diagnoses generation,
- verification and repair.

This paper is mostly concerned with systematic conflict generation. The problem of conflict generation appears to be one of the most important problems in automated diagnosis of dynamic systems based on domain model of correct system behaviour.

Conflict sets [22] (or *conflicts*, for short) are the sets of components of the system such that under the assumed model and observed output not *all* the components of any conflict set cannot be claimed to work correctly. Such sets of "suspected" elements are used then for potential diagnoses generation in the form of hitting sets for all the conflicts (i.e. a diagnosis is any set having nonempty intersection with any conflict set, and build from the elements of conflict sets only). This kind of diagnostic approach is based on Reiter's theory [22] of diagnosis from first principles, and DeKleer's work on diagnostic systems [5]. In application to dynamic systems the theory describing system behaviour is constituted by a causal qualitative model of correct system behaviour in the form of CA-EN causal graphs incorporating qualitative calculus [2].

When considering the problem of conflict set generation, the following simplifying assumptions will be considered to hold:

- the causal qualitative model is complete in the sense that the behaviour of all variables of interest can be effectively calculated (simulated) under the assumption of correct behaviour of system components,
- the possibly observed incorrect behaviour of certain variables is due to one or more faults of components only; no misbehaviour caused by incorrect design or implementation, closed-loops feedback effects, wrong control actions (input variables), external noise, impreciseness of the model or measurements are taken into account,
- the potential faults can be caused only by elements "assigned" to influence relations represented by edges of the graph; for simplicity one can assume that one influence is represented by one "identifiable component" assigned to it,
- quasi-static faults are considered only, i.e. faults that can be observed for some time period, causing steady-state-like misbehaviour observed during some time interval; no temporal misbehaviour is considered here, i.e. faults are assumed to be of permanent nature,
- the structure of the subgraph of interest does not change during its analysis,
- the considered graph has no loops,
- no time (dynamics) is taken into account,
- all the influences are "calculable", i.e. the equations describing the signal propagation are solvable both in the forward and in the backward direction,
- all the misbehaving variables can be detected.

Further, no knowledge about potential misbehaviour modes of the components will be used. The assumed goal is to find, possibly all, explanations, i.e. faults, responsible for the observed misbehaviour.

2 Graphical notation

For the sake of representing graphically various features concerning analyzed cases a simple extension of the causal graphs symbolics is proposed. Throughout the paper the following extension of the basic notation of [2] will be used:

- P, Q, R – input/control variables,
- U, V, W – intermediate variables,
- X, Y, Z – output variables; X is also used as any variable (without making precise its position),
- $[U]$, [ ] – a variable not measurable,
- X, $\underline{X}$, $\bullet$ – a measured variable,
- X , X^*, $*$ – a variable observed to misbehave,
- $X+$, X^+, $X-$, X^- – extended notions of misbehaviour, i.e. providing the information that the value is too big or too small, respectively (reserved for future use).

As usually, an arrow ($\longrightarrow$) means causality. Families of variables are to be denoted with boldface characters. e.g $\mathbf{X}$ = $\{X_1, X_2, \ldots, X_k\}$. Influences (equations) are denoted by I, e.g. $P \xrightarrow{I} X$ means that P influences X through I; a component responsible for the correct work of I is to be denoted by c. Faulty components or influences will be also denoted by c^* and $c*$, respectively. For simplicity, assuming that one component c is responsible for the correct behaviour of influence I, we can interchange components with influences and vice versa.

In case of dynamic equations a “time-flattening” procedure may be applied. This means that a differential equation can be replaced with an appropriate set of algebraic equations describing the relationship among the variable values in the subsequent time instants, as it is done for numerical solution of differential equations.

3 Causal graph

The class of considered causal graphs is quite a general one. By a *causal graph* we understand a set of variables (taking either numerical or symbolic values) and represented by the graph nodes, and a set of *causal influences* defined with appropriate equations, and represented by the arcs of the causal graph. Thus any causal graph is assumed to be a structure of the form **G** = $(\mathbf{X}, \mathbf{\Psi})$, where **X** is the set of all the variables and $\mathbf{\Psi}$ is the set of influences/equations allowing for calculation of certain non-input variables on the base of the input ones. It is assumed that the equations defined by $\mathbf{\Psi}$ are forward and backward calculable, i.e. having all input variables for one influence the output can be calculated, and having the all but one input variable values and the value of the output variable, the single undefined input variable can be calculated. It is no matter here if the calculation are analytical or numerical ones.

4 Basic problem formulation

A reasonable assumption is to start the diagnostic procedure at discovering that at least one of the observed (i.e. measurable and measured) variables misbehaves. This can be done basically in two ways:

- the detection can be a *model-based* one, i.e. the value of the variable is predicted (by simulation, under assumption of correct work of all the components) and compared with the observed value. In case significant discrepancy is observed, the variable is classified as misbehaving or non O.K. This method is in fact applied in CA-EN [2]; some further details considered there cover the problem of minimal time period of the discrepancy being observed (to avoid false alarms caused by fluctuations, etc.), and problems following from qualitative character of the values of the considered variables.

- the detection can be based on *expected behaviour* approach [12, 11], i.e. the definition of *expected normal behaviour* or *expected failure behaviour* can be stated explicitly. The definitions can be based both on the analysis of the model and on the expert domain knowledge and former experience. The advantage of the latter approach is that the detection can be quicker, as it does not require simulation with use of the system model.

Moreover, any combination of the above two approaches can be applied. In case of complex systems with qualitative models and big degree of uncertainty and vagueness in the domain knowledge, such a combination may be necessary in order to reasonably cover most of the real failures and avoid covering the false ones. Further, general “integrity constraint”, e.g. in the form of logical formulae can be provided to describe consistent and inconsistent patterns of variable/values combinations [14].

The starting point for diagnostic procedure consists in determining a nonempty set of misbehaving variables $\mathbf{X}^* = \{X_1^*, X_2^*, \ldots, X_k^*\}$. We assume that during the operation of diagnostic procedure this set remains unchanged, i.e. quasi-static faults are to be diagnosed. Our goal here is to determine possibly all minimal conflict sets for the observed set $\mathbf{X}^*$. This conflict sets may be used then for diagnoses generation.

It seems reasonable to distinguish the following stages to be carried out in course of a systematic conflict generation procedure:

1. **Domain restriction:** Restriction (possibly maximal) of the initial graph to a subgraph containing only the variables and components “involved” in the creation of observed misbehaviour; however, certain *boundary* variables may also be useful, e.g. to eliminate certain suspected components and/or to further structure the potential conflict sets. Note that in a more complex system several independent faults may occur at a time in “geographically separated” areas of the system; it seems reasonable to perform diagnostic reasoning then independently for any such area,

2. **Strategy selection:** Establishing a strategy for conflict generation, e.g. “hot” starting points, order of generation, restrictions, etc. One of the key issues there is that the generation of conflicts should be efficient both with respect to time of generation and with respect to their “parameters” (conflict sets should be as precise as possible, i.e. minimal),

3. **Efficient conflict generation:** Systematic conflict generation, usually from from the "smallest" to the "biggest" ones with deleting non-minimal conflicts and efficient elimination of potential conflicts sets which are not real conflicts. This point is crucial for the diagnostic efficiency – if not all the conflicts are generated, some faulty elements may be missing in final diagnoses; if conflicts are not minimal, too many diagnoses are likely to be obtained.

The conflict generation stage can be interleaved with diagnoses generation. A post-analysis of conflicts after generation stage may be useful as well; elimination of certain conflicts – if possible – leads to smaller diagnoses, while considering minimal conflicts leads to smaller number of potential diagnoses. In order to establish efficient strategy both *preferences* and current *limitations* should be taken into account. Further, expert domain knowledge and heuristics may be useful (e.g. in the form of pre-schedules or plans for ordering conflict generation).

Below we consider the three above stages as separate problems.

5 Graph restriction

Let us consider the problem of restricting the area of focus in order to minimize the domain for potential conflict calculation. This can be done by restricting the initial graph **G** to a subgraph, say **G'**, sufficient for the diagnostic task. Intuitively, the goal is to rule out most of the variables and influences not taking part in the formation of the observed misbehaviour. We discuss below some of the most straightforward possibilities.

Consider the most simple and intuitive restriction consisting in limiting the area of interest to elements from which there is a signal flow to the misbehaving variables (as in [2]); in other words, the idea of causality is applied to elimination of items not having the causal influence on the observed misbehaviour. Let $ANT(X)$ denote the subgraph composed of components (influences) such that through any c of $ANT(X)$ there is a directed path to X. Similarly, let $DESC(X)$ denote the subgraph composed of all components (influences) such that there is a directed path from X through any $c \in DESC(X)$. Similar notation can be applied to sets of variables. Only the elements involved in $ANT(X)$ can have influence on the behaviour of X.

It would seem natural to limit the area of interest to $ANT(\mathbf{X}^*)$ but the simplest example concerning back-propagation shows that in some cases it may be not sufficient (see Fig.1).

Note that, in such a case at least two problems following from missing of some auxiliary information occurs:

- lack of intermediate point checking,
- compensation phenomenon for multiple faults in line.

In case c_1 is faulty c_3 may compensate for its misbehaviour so that Y behaves correctly. In such a case even if $c_3 \notin ANT(X^*)$, it should certainly be taken into account. The use of it may be twofold: if $\{c_2, c_3\}$ is a conflict set (apart from $\{c_1, c_2\}$), then probably c_2 is faulty; however the explanation that c_1 and c_3 are faulty and c_3 compensates for the fault of c_1 at Y is also possible (the compensation phenomenon takes place). In case $\{c_2, c_3\}$ is not a conflict set, then most probably c_1 is the only faulty element there. Thus the part with c_3 provides new discrimination information. We shall refer to this type of structures as "fork-like".

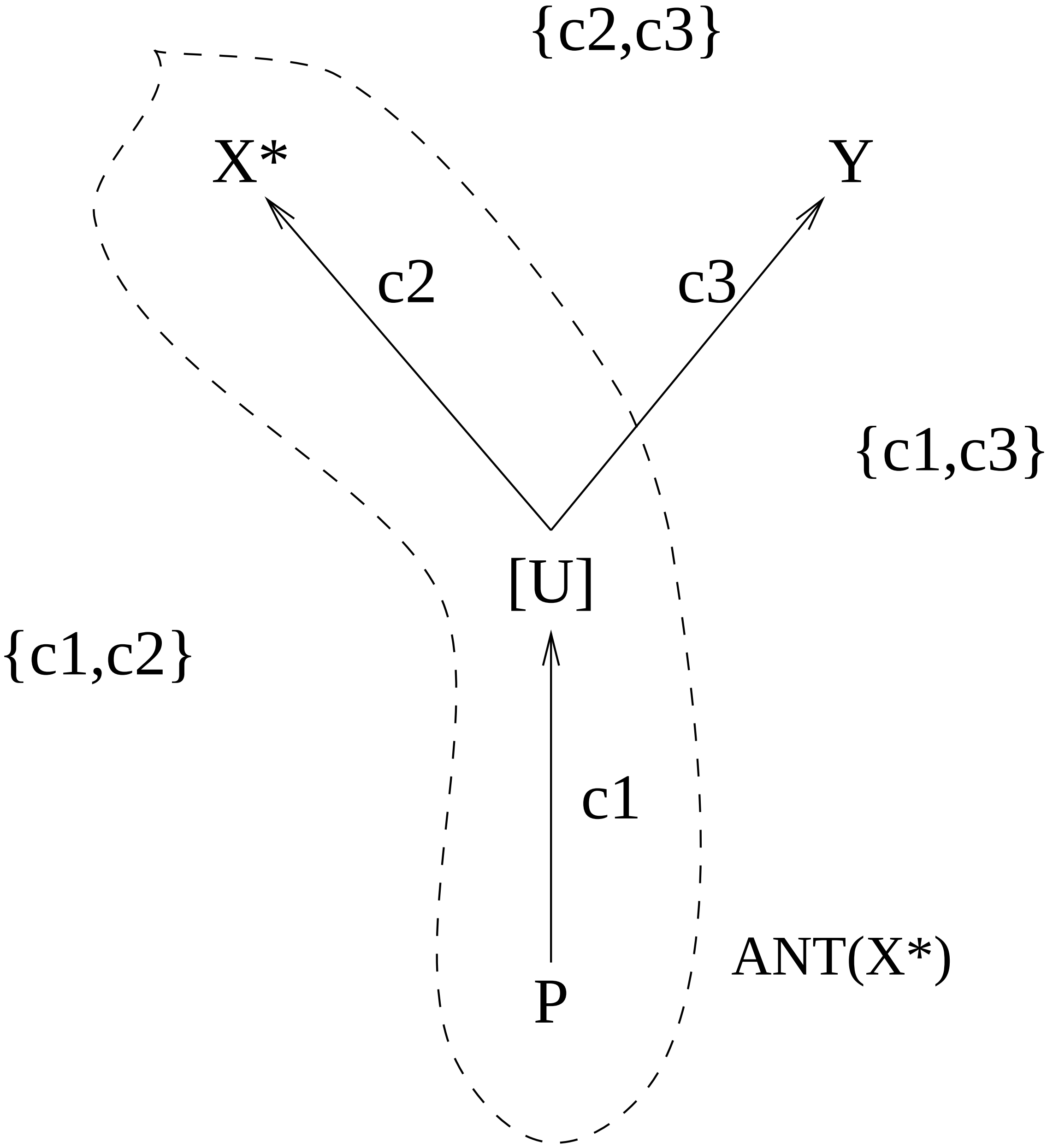


Figure 1: Example – restriction of the domain to $ANT(\mathbf{X}^{*})$.

The next approximation may be to limit the area of analysis to the set defined as $DESC(ANT(\mathbf{X}^{*}))$ and it seems to be quite reasonable[1]. However, again, extension of the former recent example shows that in certain cases this heuristic simplification may lead to incomplete conflict generation possibilities (see Fig. 2).

One of the possible conflict sets is $\{c_3, c_8, c_7, c_6, c_5, c_4\}$, and in order to calculate it one has to consider a complex combination of ascending/descending elements. Of course this example may be extended horizontally on any arbitrary number of variables; we shall refer to it as "side-

[1] This, in fact, is the core of the approach equivalent to assuming that back-propagation is not combined with propagation forward.

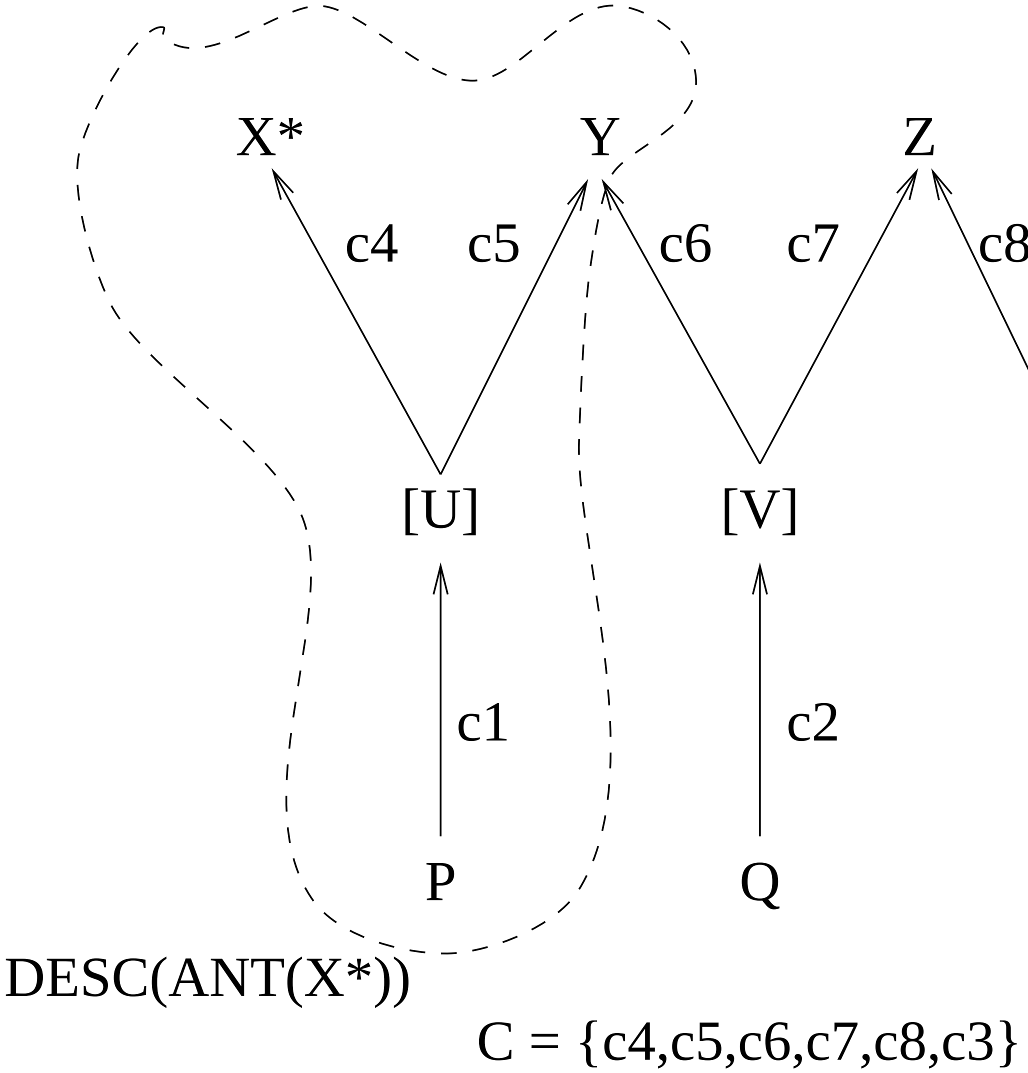


Figure 2: Example – Restriction of the domain to $DESC(ANT(\mathbf{X}^*))$.

wave" or "side-effect" example.

The next obvious possibility is an arbitrary combination of the form

$$DESC(ANT(\ldots DESC(ANT(\mathbf{X}^*))\ldots))$$

or similar; the main problem is where to stop. Of course, an obvious solution is to cover all the connected graph, i.e. break the procedure of subgraph growing on the "natural boundaries". Below a more restrictive proposal, still satisfying the requirement of generating all possible conflicts is outlined.

Let us consider an arbitrary subgraph of the initial causal graph containing at least one unmeasured variable. By *extending* the graph we shall understand adding subsequent links (with assigned to them variables). Extending on a certain path (paths) to or from an unmeasured variable leads to a *closure* if all the "boundary" variables are measured ones, and the values of the unmeasured variables incorporated in the subgraph can be calculated (either by for- or by back-propagation) from the boundary variables. The smallest set of measured boundary variables defining a closure for the variables of the initial graph on all the paths to or from it "cuts out" the subgraph of interest. It is to be denoted by $CLO(\mathbf{X})$ for the initial set of variables being $\mathbf{X}$. In other words, the $CLO(\mathbf{X})$ is a minimal subgraph covering $\mathbf{X}$, "cut off" from the basic causal graph at measured variables only (and including all of them). Below we propose a formal definition following the above intuitions.

Definition 1. *Let $\mathbf{X}$ be an arbitrary set of variables (both measured and unmeasurable ones). The closure of $\mathbf{X}$, to be denoted as $CLO(\mathbf{X})$ is a subgraph of the causal graph satisfying the following conditions:*

- *$CLO(\mathbf{X})$ incorporates all the variables of $\mathbf{X}$,*
- *all the input and output variables of $CLO(\mathbf{X})$ are measured ones,*
- *all the input and output variables are the O.K. ones,*
- *$CLO(\mathbf{X})$ is minimal with respect to set inclusion of the set of subgraph nodes.*

The meaning of input and output variables is straightforward. An input variable is one from which the signal is directed inside the structure and taking it value from outside the structure. An output variable is one taking its value from inside the graph and, if some links point from this variable they must all go outside the closure. Thus a variable to which several links point can be a boundary variable only if all the links are pointing inside or outside the closure. For example, on Fig. 2 variable $\underline{Y}$ is not a boundary variable for the structure incorporating elements c_1, c_4, and c_5.

For intuition, the construction of the $CLO(\mathbf{X})$ cuts out a specific subgraph from the initial graph, but one can cut only through measured and correct variables and not through links (see Fig. 3).

This means that one should try to isolate the subgraph with respect to information coming in or out of the subgraph; one tries to isolate it from information theory point of view. Of course, the selected subgraph should be the smallest one, covering the misbehaving variables. Further, the variables through which we cut should behave correctly - the goal is to isolate misbehaving part of the graph. This is also like growing the misbehaving area until correctly behaving measured variables are reached on all paths going outside.

The input and output variables will be referred to as *boundary variables*; in fact, they constitute a kind of a frontier such that all the information coming in or out goes through the boundary variables. Intuitively, if they behave O.K. no information about misbehaviour inside $CLO(\mathbf{X})$ can go out from the structure and thus be manifested outside; similarly, no information about possible faults outside can go inside the structure and thus be a cause for the observed misbehaviour of the incorporated variables.

Note that the above operation separates in fact a subgraph closed from the point of view of information theory; assuming the "Markow-like " character of the causal graphs, no information can be transferred inside or outside the CLO-sured subgraph in a way different than through the boundary variables. Thus, no information from the outside can have influence on the diagnostic process, provided that the goal is to explain the misbehaviour of the variables inside the CLO-sure.

Now the natural consequence of the above idea is to restrict the subgraph for conflict generation to be $CLO(\mathbf{X}^*)$ (see Fig. 3 for and intuitive idea). Moreover, if several separated subgraphs

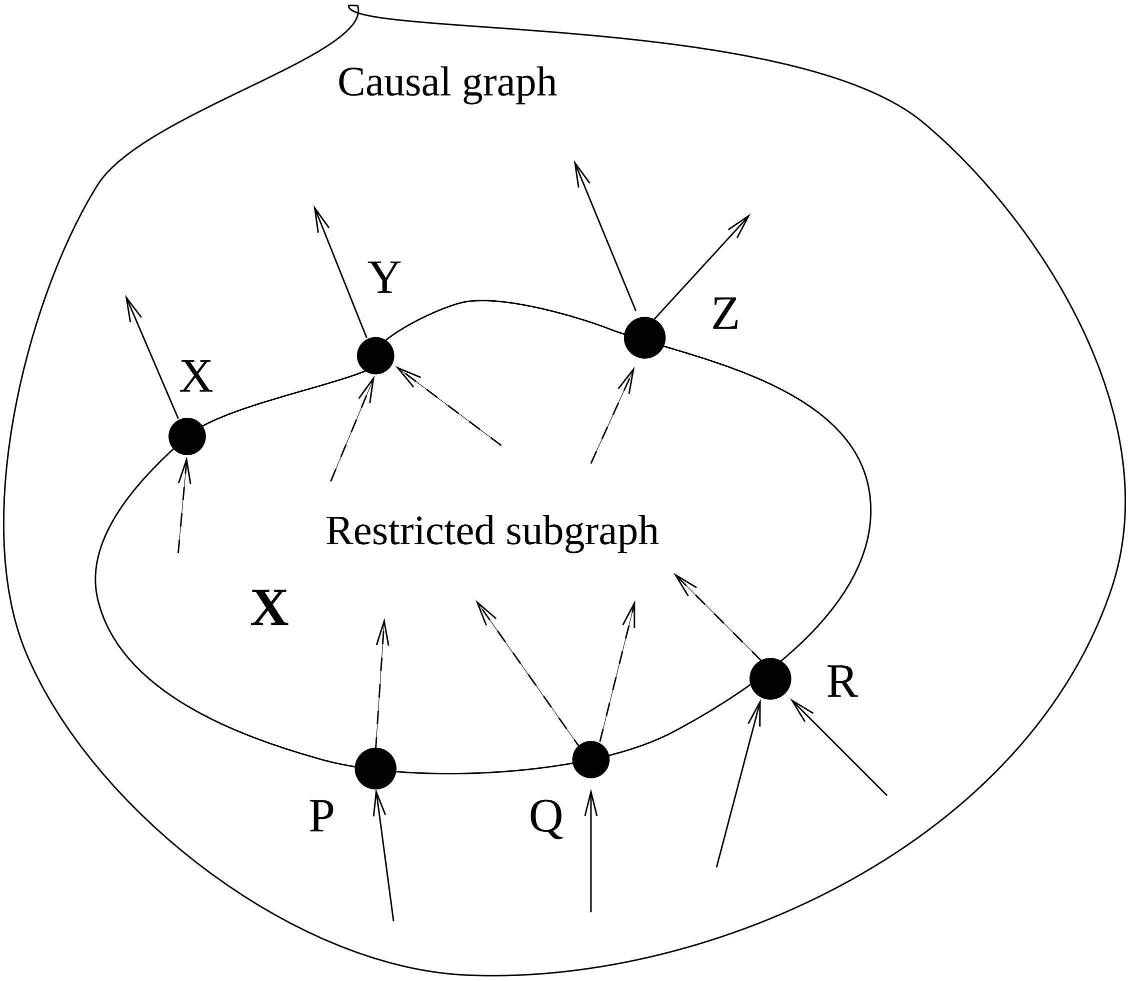


Figure 3: An information closure.

consisting closures for the misbehaving variables can be constructed, the diagnostic process can be performed for any identified sub-area independently from each other. Roughly speaking, in such a case the closure would constitute “islands”of isolated misbehaviour on the area of the initial causal graph.

The idea of the algorithm for constructing $CLO(\mathbf{X}^*)$ may be as follows. First construct a $CLO(\{X^*\})$ for any variable $X^* \in \mathbf{X}^*$. This can be done by following any path to and from X^* until a measurable and correct variable is spotted. Then any closures of single variables sets having some elements in common should be composed into connected subgraphs. The process may result with one or more connected subgraphs.

Note that the isolation process can be done for both misbehaving and O.K. variables, i.e. one can perform a construction like the above closure for a set of correct variables. In this case, such a closure should be excluded from the diagnostic process (maybe without the boundary variables).

As it can be seen, the above choice is quite a natural one, and it follows also from the approach to conflict calculation presented later.

6 Strategy for conflict calculation

In order to consider the problems of strategy of conflicts generation, one should first answer the questions concerning preferences among diagnoses to be generated and sets of diagnoses viewed as "final solutions ".

Recall that any diagnosis D is just a set of components, $D = \{c_1, c_2, \ldots, c_m\}$ such that assuming them faulty is sufficient for restoring the consistency of the domain theory with the observations. The sets of diagnoses are denoted as $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$..

Considering preferences among diagnoses we must take into account the risk of basing our diagnostic procedure on incomplete set of conflicts, i.e. a case when not *all* the conflicts are calculated. Intuitively, the diagnoses calculated in such a case will be "incomplete" or "partial". This observation is supported by the following proposition.

Proposition 1. *Let $\mathbf{C}_i$ denote sets of conflict sets, and $\mathbf{D}_i$ sets of diagnoses calculable from $\mathbf{C}_i$, $i = 1, 2$. Then, if $\mathbf{C}_1 \subseteq \mathbf{C}_2$, then also:*

1. $||\mathbf{D}_1|| \leq ||\mathbf{D}_2||$, *and*
2. $\forall D_1 \in \mathbf{D}_1 \; \exists D_2 \in \mathbf{D}_2$ *such that* $D_2 \supseteq D_1$.

According to the above proposition, in case of more conflict sets accessible (i.e. known; in our case the problem is to discover the conflicts since they are "hidden" in the graph structure), the diagnoses are possibly more precise[2]. Another observation is that simultaneously, having more conflicts we can expect generation of more diagnoses, and this is what we would like to avoid. As the diagnoses are only potential explanations of the observed misbehaviour, they require further verification, thus the number of diagnoses generated should be relatively small. The intuition is that generation of the conflicts should be limited to the minimal ones (or at least as small as possible). The following proposition is related to the problem of limiting the number of possible diagnoses.

Proposition 2. *Let $\mathbf{C}_i$ denote sets of conflict sets, and $\mathbf{D}_i$ sets of diagnoses calculable from $\mathbf{C}_i$, $i = 1, 2$. Further, assume that $\mathbf{C}_1$ and $\mathbf{C}_2$ have the same number of elements. Then, if for any $C_1 \in \mathbf{C}_1$ there exists $C_2 \in \mathbf{C}_2$ such that $C_1 \subseteq C_2$, then there is also $||\mathbf{D}_1|| \leq ||\mathbf{D}_2||$.*

The above proposition allows for comparison of two sets of conflicts with the same number of elements, but different "degree of preciseness"; the more precise conflicts are better, they lead to generating less possible diagnoses.

Finally, let us consider the problem of adding new conflict sets to an existing set of conflicts. If in the already found conflict set there is one smaller than the one to be added, then adding the new one is not necessary; either no new diagnoses will be generated or the diagnoses will not be minimal. The following proposition states it more precisely.

[2] At this stage we assume to prefer as precise (complete) diagnoses as possible. By a more precise diagnosis we understand here one explaining more of the observed failures; contrary to precise (complete) diagnoses, the imprecise (incomplete) ones explain only part of the observed misbehaviour. This may seem contrary to the usual preference of minimal diagnoses. Our standpoint is, however, following from the risk that not all the conflicts are likely to be generated. For an established set of conflicts preferring minimal diagnoses is still an obvious choice.

Proposition 3. *Let $\mathbf{C}_i$ denote sets of conflict sets, and $\mathbf{D}_i$ sets of diagnoses calculable from $\mathbf{C}_i$, $i = 1, 2$. Let $\mathbf{C}_2$ contain all the conflicts of $\mathbf{C}_1$ and some other not minimal ones, i.e. $\mathbf{C}_2 = \mathbf{C}_1 \cup \mathbf{C}_1'$, where for any $C_1' \in \mathbf{C}_1'$ there exists $C_1 \in \mathbf{C}_1$ such that $C_1 \subseteq C_1'$. Then for any diagnosis $D_2 \in \mathbf{D}_2$ there is a diagnosis $D_1 \in \mathbf{D}_1$ such that $D_1 \subseteq D_2$. Further, there is also* $||\mathbf{D}_1|| \leq ||\mathbf{D}_2||$.

This proposition justifies the intuition that non-minimal conflicts are useless for diagnostic efficiency – adding non minimal conflicts not only may lead to more diagnoses, but also to generation of non minimal ones as well.

The above considerations seems to justify the following assumptions concerning the strategy of conflict generation:

1. Conflicts should be generated in a systematic way, so that all the necessary conflicts are obtained; if a conflict set is missing, there is a risk of generating partial (incomplete) diagnoses.

2. Conflicts should be generated from $i = 1$ towards $i = k$, where i is the number of components in a conflict set and k is the maximal number of components in the analyzed subgraph; this assures that more precise conflicts are generated first.

3. All conflicts comprising i elements should be calculated before ones comprising $i + 1$ elements; a conflict which is a superset of some previously generated conflict is not to be considered.

4. The procedure can be stopped when either no new conflicts can be generated, or for any new conflict to be generated a subset conflict has already been generated (minimality requirement).

Note that the following further auxiliary rules may be proposed for enhancing he diagnostic process:

- the conflict generation procedure may be stopped for some number of conflicts generated arbitrarily; this may be the case when diagnoses containing only some limited number of faulty components are probable,

- the conflict generation procedure may also be stopped for some i arbitrarily; this may be the case when too complex conflicts are too costly to calculate, etc.,

- more data (measurements, tests) may become available cutting down in a natural way the size of conflict sets,

- conflict generation may be interleaved with diagnoses generation (see also [22]); some diagnosis found valid may stop the process,

- expert-designed *schedules* for conflict generation finding the most probable conflicts "around" elements most likely exhibiting faulty behaviour can be used to speed-up the diagnostic procedure by turning it into routine procedures,

- if accessible, the knowledge about modes of faulty behaviour of the components and its influence on the behaviour of variables can be used for further selection.

7 An approach to systematic conflict generation

The basic assumption here is that both propagation and back-propagation can be regarded as *mathematical constraints*, i.e. they provide some equations determining the relations among variable values. An equation with one unknown value of a variable defines this value; if all the values are known, then such an equation provides a possibility of conflict generation – the components responsible for holding of this equation may not be all working correct if the equation is not satisfied by the observed variable values.

In case of back-propagation there can be some difficulties with "inverting" the calculations so as to obtain the values for one of the arguments. However, from theoretical point of view it seems that in any case one can solve the equation numerically, and so inverting it is not a necessary procedure; of course, if applicable, it can contribute to computational efficiency.

From purely mathematical point of view, in order to generate conflicts one must have more equations than variables; in our case they are unmeasured variables. Thus if n denotes the number of equations (both for back- and for-propagation) defined for some subgraph, and m is the number of unmeasured variables involved in the computation, then the condition necessary for potential conflict generation is that $n \geq m + 1$. Further, for any such substructure there exists a potential possibility of generating no more than $\binom{n}{m+1}$ possible conflicts. This can be illustrated with Fig.4, where $n = 4$, $m = 1$, and we have $\binom{4}{2}$ potential conflicts.

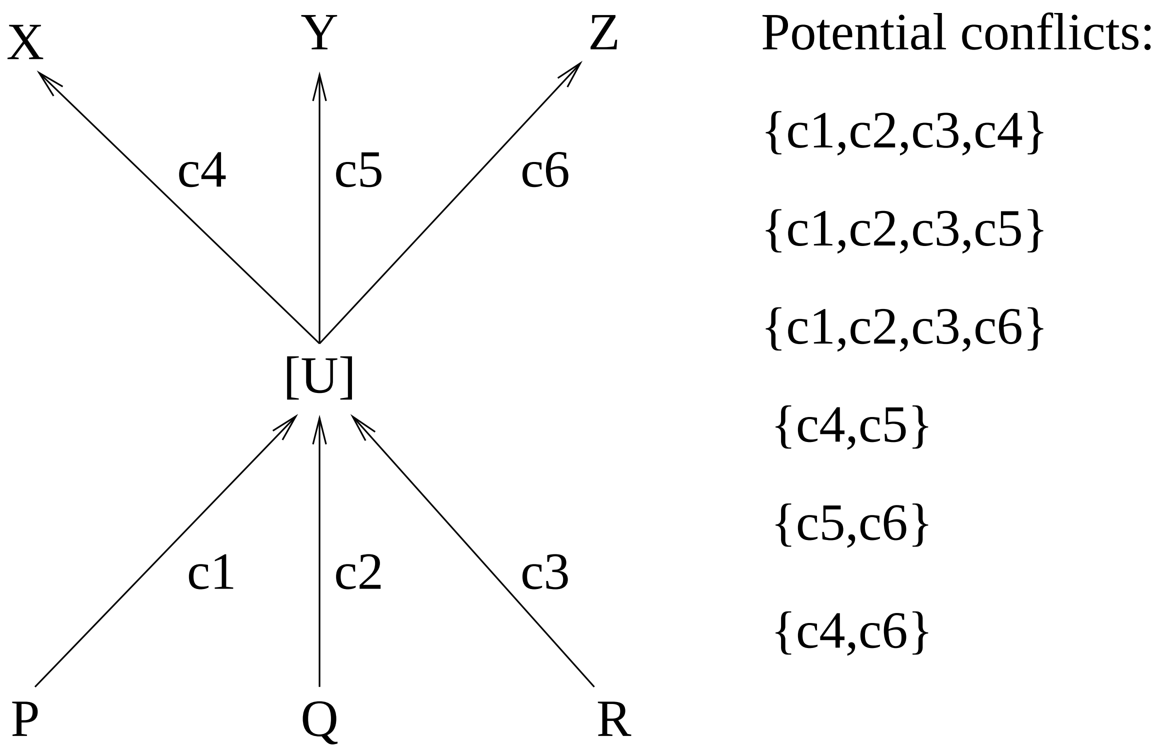


Figure 4: Example – potential conflict structures selection.

Usually, there are less conflicts, since not all the structures described by n equations and containing m variables allow for calculation of a "full-size" conflict; examples include chains with pending unmeasured variables. Further, all the conflicts are only *potential* – if a conflict is really observed or not depends on actual computations, and, of course, on the existence of faulty elements (it is assumed that no conflicts are generated due to inadequate calculations or inadequate model).

Taking into account the above considerations and in order to achieve better efficiency of conflict generation, the following, two-stage, transparent procedure of conflict generation is proposed;

- identification of *potential conflict structures*, i.e. sets of influences assuring *necessary* conditions for conflict existence, and then
- verification for any such a structure and selected set of equations if a conflict exists; this is to be done by an attempt at "solving" these equations.

Splitting the procedure of conflict generation into these two stages seems to advantageous for the sake of transparency and systematic conflict generation. Moreover, identification of a conflict structure is equivalent to having the knowledge about it components. Thus exploration of non-minimal potential conflicts can be abandoned without performing the real calculations. Moreover, certain heuristics can be applied to preselect the potential conflict structures for further investigation, leaving a large part of them without performing costly mathematical calculations.

The key issue for carrying on is to introduce a definition of a *Potential Conflict Structure*, shortly *PCS*. This notion denotes a subgraph of the causal graphs, for which there is a possibility (always potential) of calculating a conflict via obtaining two different values for the same variable. A PCS comprising m unmeasured variables and leading to detection of potential conflict at a variable X will be denoted as $PCS_m(X)$. The number of unmeasured variables m will be referred to as the *order* of a conflict structure. Variable X can be measured one or an unmeasured one.

Note that some most interesting are potential conflict structures having no unmeasured variables, i.e. PCS_0 – they are always of the form $P_1, P_2, \ldots, P_j \xrightarrow{c} X$, where all the variables are measured; if such a structure provides a real conflict, then the conflict consists of one element c and in fact is a partial diagnosis. In other words, component c is faulty and must be an element of any valid diagnosis; further the fault of c is a cause of the observed misbehaviour of X. Therefore conflict structures of zero order should always be explored first (if existing).

Now let us pass to potential conflict structures of larger size. First we put forward the following definition.

Definition 2. *A variable X is well-defined (defined, for short) iff:*

- *either it is a measured variable, or*
- *its value can be calculated on the base of some other variables, which are well-defined.*

If there are two or more, e.g. k independent ways of calculating the value of X, then X is said to be k-defined.

The independent ways of calculating the variable may consist in measuring the value and calculating it with different sets of equations. The calculation of a well-defined variable can be done no matter how – forward propagation is as good as backward one (at least from purely mathematical point of view). Now we can define a potential conflict structure on m unmeasured variables.

Definition 3. *A* **<u>Potential Conflict Structure</u>** *for variable X defined on m unmeasured variables is any subgraph of the causal graph, such that:*

- *it comprises exactly m unmeasured variables (including X, if unmeasured),*
- *all the variables are well-defined, and X is double-defined;*
- *for the PCS being defined on m unmeasured variables it is necessary that all the values of the m variables are necessary for making X double-defined.*

Variable X will be called the head *of the PCS.*

Examples of PCS_m for $m \in \{0, 1, 2\}$ are shown on Fig. 5.

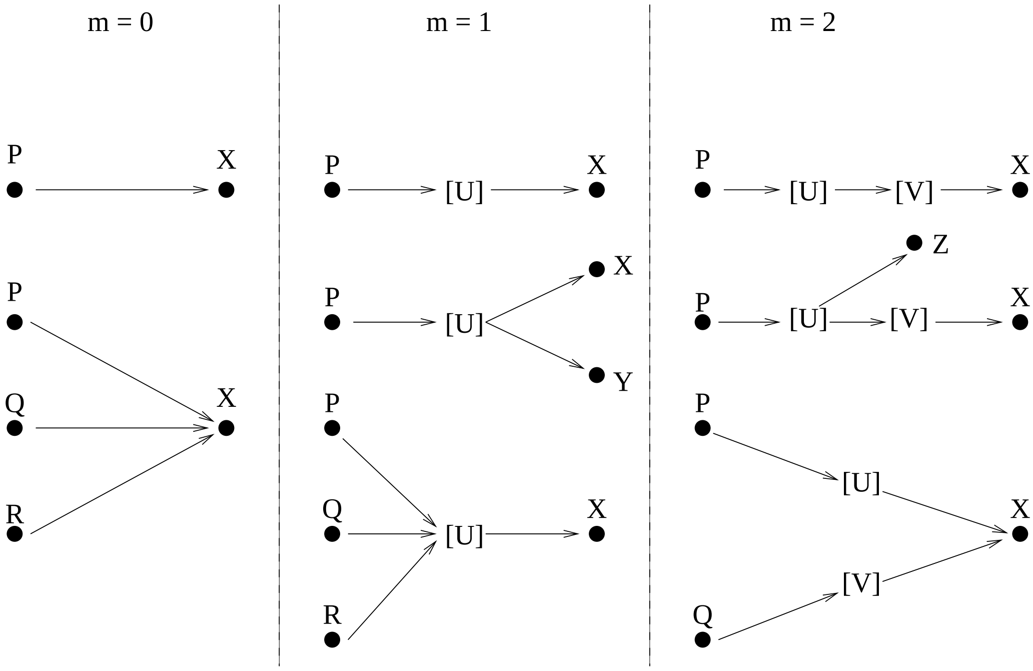


Figure 5: Examples – potential conflict structures.

Note that any of the graphs represent at least two potential conflict structures, i.e. ones for different head variable; in these cases the PCS constitute the same graphs and the same conflicts will eventually be generated. Thus for any such PCS the calculation is to be performed only once, and the selection of the head variable is to be done arbitrarily, e.g. with respect to making easier the problem of equations solving.

Potential conflict structures can take arbitrary "shape" and it is, in general, difficult to say if some structure is a PCS at the first sight. The definition is in fact recursive, and so the algorithm for detection of PCS must be. But there are at least three typical conflict structures with some nice properties and interpretation. They are: a "chain" of calculable variables, a "pyramid" and various types of "forks ", where back-propagation plays important role.

The algorithm for detecting a PCS on m unmeasured variables is a recursive one; the starting point is a selected variable, the head of the structure; then, it must search for exactly m unmeasured variables connected to the selected one if it is a measured variable (a kind of path tracing), or $m - 1$ unmeasured variables if the head is an unmeasured variable itself. When traversing the graph a check if all the unmeasured variables are well defined and if having them the head variable is double defined is to be performed. The basic procedure should be repeated for any selected variable for m changing from 0 to the maximal number of unmeasured variables.

Note that the basic procedure does not guarantee that the conflict sets are generated according to growing number of elements; this is so, because there can be many influences “pointing to” a single unmeasured variable and thus a potential conflict set may contain many elements assigned to these influences. However, for any PCS_m it is possible to assign a number j of elements assigned to the influences necessary for conflict calculation; this can be done before the calculation of the appropriate conflict; there is also always $j \geq m+1$. Thus there is a simple way of defining a second characteristics of any PCS, i.e. a number j of elements of the conflict set to be possibly determined; this may be noted as PCS^j_m. And finally, during calculation of conflicts for the same m for any unmeasured variable, the order of calculations can be done with respect to the value of j.

8 An outline of algorithmic approach

To summarize, an outline of an algorithm for systematic conflict generation can be as follows:

1. Detect the set of misbehaving variables $\mathbf{X}^*$,

2. Define the restricted subgraph being the object of analysis to be $CLO(\mathbf{X}^*)$; any other choice (e.g. a heuristic one) is possible, but the completeness can be violated,

3. For any measured variable $X \in CLO(\mathbf{X}^*)$ detect all the conflicts calculable without the use of the values of unmeasured variables; the calculation of conflicts can be ordered with respect to the number of elements in conflict sets; this step refers to the zero-order conflict generation,

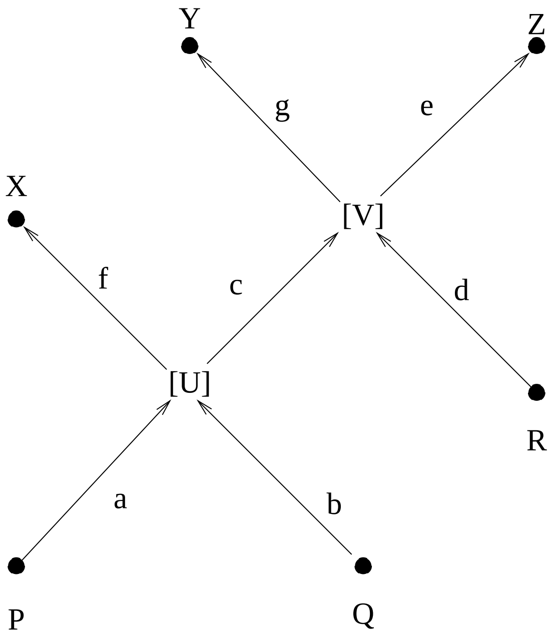
4. For any variable $X \in CLO(\mathbf{X}^*)$ detect sequentially all PCS^j_m; the order of generation is from $m = 1$ to the number of variables in the subgraph of interest. For any variable and m established order the PCS^j_m according to increasing values of j,

5. Repeated PCS, i.e. ones different only with respect to the head variable should be abandoned (leaving exactly one of them for investigation); further, PCS leading to non-minimal conflicts can be abandoned before numerical investigation,

6. Stop the procedure when there are no more PCS to be generated (or earlier, according to some heuristics or when an appropriated diagnosis is generated).

9 An Example

A simple test program for support of direct determining potential conflict structures was implemented in PROLOG. The program is a meta-interpreter using several simple recursive rules. It calculates the potential conflict structures for a specified unmeasured variable.



m = 0	-	no conflicts
m = 1;	j = 2	{e,g}
	j = 3	{a,b,f}
m = 2;	j = 4	{f,c,d,g}
		{f,c,d,e}
	j = 5	{a,b,c,d,e}
		{a,b,c,d,g}

Figure 6: Example – generation of PCS-s.

An example subgraph CLO-sured with measured variables is shown in Fig. 6; potential conflicts calculated with use of the program are listed there as well.

During calculation of the conflicts the following rules may be useful in order to avoid repeated computations and improve the overall efficiency:

- *head variables preselection:* this seems to be a most important one heuristic rule for achieving reasonable efficiency; the candidate variables for heads of the PCS should be preselected. A reasonable heuristic may consists in selecting all the misbehaving variables and the unmeasured ones (other variables, i.e. the measured O.K. ones, will be incorporated in the calculated PCS or simply are not necessary; a strong simplification may consist in selecting the non O.K. variables as heads for PCS-s,

- *eliminating repetitions:* the concept of PCS allows for identification of conflict set elements before calculation of the potential conflict; thus, whenever a PCS identical to another already generated appears, there is no need to calculate the conflict once more. This is important, since starting from different variables and extending the PCS-s it seem unavoidable to generate the same PCS several times.

- *limiting the size of generated conflicts:* again, if a PCS has been generated such that its elements constitute a superset of a conflict set which has already been generated the there is no need to investigate this PCS,

- *further decomposition:* generation of conflicts can be stopped at a boundary composed of measured variables; this is similar to considering the CLO-sure, but this time for a substructure of the selected subgraph,

- *re-use of calculations:* once calculated, influences (values of the variables) can be reused in calculation of several conflicts; they need just to be stored,

- *user-defined scenarios:* some *schedules* or expert-defined *scenarios* providing guidance for calculating conflicts in specific cases defined by the selection of misbehaving variables and type of their misbehaviour can be used to guide the procedure of conflict generation,

- *measurement introduction:* for certain PCS seeming too large, suggestions of measurement points can be done so as to structure them down into manageable objects.

- *pre-elimination of potential faulty elements:* the observed nature of faults may indicate that certain types of faults are not to be taken into account; this observation may lead to eliminate certain components from further considerations. Hence, even if the set of influences used for conflict generation are large, the generated conflicts may turn out to be quite small.

10 Closing remarks

From the above considerations one can expect that the final efficiency of conflict (not diagnoses!) generation is bound to depend on a variety of factors. Further, one can expect that efficient conflicts, i.e. ones leading to a small number of well-localized diagnoses can be obtained only if the unmeasured variables are relatively sparse. In case there are only few measurements one cannot expect that the isolation of faulty components will be effective.

With respect to this problem, the back-propagation seems to appear to be a crucial issue – roughly speaking, it may play a role similar to introducing measurements and thus contribute to limiting the size of conflicts.

Another aspect which seems well worth investigating is the problem of modes of faulty behaviour of the components and their influence on the observed behaviour of variables; an a priori knowledge about possible faults of components can be used to model the faulty behaviour and thus to select out certain possible faults if the modeled behaviour is not observed. Further, certain conflict sets can be eliminated during generation.

But the most challenging issue seems to be formalization of an approach based on combination of direct search for faulty components (as in [15]) with procedures based on conflict generation; Let us recall that the approach based on conflict generation according to Reiter's theory [22] is justified only under several relatively strong assumptions; further, after generating conflicts a next stage for generating diagnoses is necessary. The overall procedure seems to be hardly fitting on-line systems requirements (not to say about the real-time ones). On the other hand, the idea of such a backward-search procedure seems to follow from the general search principles and abductive reasoning: by appropriate use of functional element descriptions and information about measured values one should construct the hypotheses explaining the observed behaviour. In order to consider some more specific bases for such an approach, an "axiomatization" of the domain seem to be necessary. Then the basic step should consist in generation of a search space for abductive diagnostic reasoning, providing a model fitting the diagnostic purposes.

Acknowledgment: The Author thanks Dr. Louise Travé-Massuyès for many comments and discussions helpful in improving this work.

References

[1] P. Baroni, G. Guida and S. Mussi, 'Causal reasoning under uncertainty with Q/C-E networks: a case study on preventive diagnosis of power transformers', in: G. Rzevski, R.A. Adey, and C. Tasso (Eds.) *Applications of Artificial Intelligence in Engineering X*, Computational Mechanics Publications, Southampton, Boston, 119-128, 1995.

[2] Bousson, K., L. Travé-Massuyès and L. Zimmer, *Causal Model-Based Diagnosis of Dynamic Systems*.

[3] L. Console, D.T. Dupré and P. Torasso: 'A theory of diagnosis for incomplete causal models', *Proceedings IJCAI'89*, 1311-1317.

[4] Console, L. and P. Torasso, An approach to the compilation of operational knowledge from causal models. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-22, No.4 July/August 1992, 772-789.

[5] J. DeKleer and B.C. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, **32**, 97-130, 1987.

[6] P. Fuster-Parra and A. Ligęza, 'Fuzzy fault evaluation in causal diagnostic reasoning', in: G. Rzevski, R.A. Adey, and C. Tasso (Eds.) *Applications of Artificial Intelligence in Engineering X*, Computational Mechanics Publications, Southampton, Boston, 137-144, 1995.

[7] P. Fuster-Parra and A. Ligęza: Diagnostic knowledge representation and reasoning with use of AND/OR/NOT causal graphs. In *International Conference on Systems Science*, vol.:III, pp. 223-230, Wrocław, Poland, September 1995.

[8] Fuster-Parra, P. and Ligęza, A., 'An approach to diagnosis of dynamic systems', Elektrotechnika journal, tom 14, vol. 3 pp. 225-230, Krakow 1995. (Presented during *International Symposium on Applications of Systems Theory*, to be held in Zakopane, Poland, October 1995).

[9] P. Fuster-Parra and A. Ligęza, 'Qualitative probabilities for causal diagnostic reasoning', in: M.A. Bramer, J.L. Nealon and R. Milne (Eds.) *Proceedings of Expert Systems 95, the 15th Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems*, Cambridge, UK, 327-339, December, 1995.

[10] Guan, J. and J.H. Graham: Diagnostic reasoning with fault propagation digraph and sequential testing. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-24, No. 10, October 1994, 1552-1558.

[11] Ligęza, A. and Fuster-Parra, P.: Qualitative knowledge representation and processing for causal diagnostic reasoning. Extended reasoning modes and efficiency-related issues. *Technical Report of the Institute of Automatics AGH*, No.: 38, Kraków, 1994, 40 pp.

[12] Ligęza, A. and Fuster-Parra, P.: Automated diagnosis. An expected behaviour based approach. In *Proceedings of the 8-th International Symposium System, Modelling and Control*, Vol. 2, Published by Polish Society of Medical Informatics, Łódź, Zakopane, Poland, May 1995, pp. 7-12.

[13] A. Ligęza and P. Fuster-Parra, 'An approach to diagnosis of dynamic systems through search of AND/OR/NOT causal graphs', J. Kocijan and R. Karba (Eds.), *Preprints of the IFAC/IMACS International Workshop on Artificial Intelligence in Real-Time Control*, Bled, Slovenia, 1995, 126-131.

[14] Ligęza, A., P. Fuster Parra, and J. Aguilar-Martin: Backward search on causal logical graphs. Yet another view on diagnostic reasoning. *LAAS Report No. 96222*, 1996.

[15] Ligęza, A., P. Fuster Parra, and J. Aguilar-Martin: Causal Abduction: Backward Search on Causal Logical Graphs as a Model for Diagnostic Reasoning. *LAAS Report*, 1996 (to be printed).

[16] Lunze, J. and F. Schiller: Logic-based diagnosis utilizing the causal structure of dynamical systems. *IFAC/IFIP/IMACS Int. Symp. on AI in R-T Control*, Delft, 1992, 649-654.

[17] W. Nejdl, P. Frohlich and M. Schroeder, 'A formal framework for representing diagnosis strategies in model-based diagnosis system', *Proceedings IJCAI'95*, 1721-1727, 1995.

[18] Nicol, C., L. , J. Quevedo, Travé-Massuyès:: TIGER: Applying hybrid Technology for industrial monitoring. *LAAS Report*, No.: 96039, 1996.

[19] J. Pearl, *Heuristics. Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publ. Co., Reading MA, 1984, 1985.

[20] J. Pearl, 'Fusion, propagation, and structuring in belief networks', *Artificial Intelligence*, **29**, 241-288, 1986.

[21] D. Poole, 'Normality and faults in logic-based diagnosis', *Proceedings of IJCAI'89*, 1304-1310.

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

[23] S.T. Russell and P. Norvig, *Artificial Intelligence. A modern approach*, Prentice Hall series in Artificial Intelligence, 1995.

[24] G. Saucier, A. Ambler and M.A. Breuer (Eds.), *Knowledge Based Systems for Test and Diagnosis*, North-Holland, Amsterdam, 1989.

[25] P. Struss, Knowledge-based diagnosis – an important challenge and touchstone for AI, *Proceedings of ECAI'92*, 1992.

[26] P. Torasso and L. Console, *Diagnostic Problem Solving. Combining heuristic Approximate and Causal Reasoning,* North Oxford Academic, A Division of Kogan Page, London, 1989.