Diagnosis and Supervision: Model-Based Approaches

Marie-Odile Cordier, Philippe Dague, Yannick Pencolé and Louise Travé-Massuyès

Abstract This chapter is devoted to diagnosis and supervision. It is organized as follows: after a section dedicated to the logical formalization of model-based diagnosis, the focus is made on diagnosis of discrete event systems modeled by automata. In the last section, one presents more succinctly the works that allowed to make the bridge between the approaches proposed by the Artificial Intelligence community and those proposed by the Automatic Control community. AQ1

1 Introduction

Diagnosis consists in observing a system (often by using sensors), in detecting from these observations possible dysfunctions or mode change (from normal to abnormal) and in identifying the fault(s) they evoke. Diagnosis can be carried out in the medical field but also in the industrial field or even the environmental, economic ones, etc. The first works in Artificial Intelligence (AI) dealing with diagnosis were, in the 1980s, the expert systems based approaches, which appeared with the application to medical diagnosis and the Mycin system. These approaches relied on general on production rules whose condition part describes observable signs and symptoms and conclusion part the diagnoses they evoke. These associative approaches for diagnosis have continued and gave rise to case-based reasoning approaches (see chapter "Case-Based Reasoning, Analogy, and Interpolation" of this volume) and, for

M.-O. Cordier (✉)
IRISA, Rennes, France
e-mail: marie-odile.cordier@irisa.fr

P. Dague
LRI, Université Paris-Sud, CNRS, Orsay, France
e-mail: philippe.dague@lri.fr

Y. Pencolé · L. Travé-Massuyès
LAAS-CNRS, Toulouse, France
e-mail: ypencole@laas.fr

L. Travé-Massuyès
e-mail: louise@laas.fr

P. Marquis et al. (eds.), *A Guided Tour of Artificial Intelligence Research*,
https://doi.org/10.1007/978-3-030-06164-7_21

dynamic systems, to chronicles (or scenarios) recognition approaches, where one associates to a set of temporally constrained events the diagnostic situation to which these events correspond.

Despite their success, it has been often reproached associative approaches for coding reasoning shortcuts, left without explanations capabilities relying on the functioning of the system to be diagnosed. This motivated the introduction of model-based approaches, which rely on a description of the behavior of the supervised system, this model being possibly limited to the behavior of the so-called normal behavior of the system studied. It will be seen in the following that it is often relevant to join it, when available, fault models, describing also the behaviors resulting from the occurrence of a fault. One can distinguish between predictive models, which allow prediction of the system's behavior, in particular the values observed by the sensors, and explanatory models, which allow explanation of observations resulting from the faults that occurred.

Diagnosis problem interested a lot researchers in AI. It actually associates a modeling problem, therefore the choice of a formalism (based on logic, graphs or constraints) for behavior representation of the system studied with its uncertainty and complexity, a diagnoses characterization problem and a heuristic algorithmic problem for solving with satisfactory efficiency a task, which is most of the time NP-hard. And this field by the way influenced considerably AI research, since expert systems, default logic, fuzzy logic and non monotonic logics, constraints, causal graphs, qualitative reasoning had often their first applications in this framework. As it will seen in this chapter, this field motivated also largely researchers in the Automatic Control field who, firstly more focused on control, expanded their interest to search of the causes of the dysfunctions detected.

It can be finally noticed that diagnosis is not in general an end per se and that the issue is to "repair" the monitored system, which relates it directly to research in decision theory (see chapters "Multicriteria Decision Making" and "Decision Under Uncertainty" of this volume) and in planning (see chapter "Planning in Artificial Intelligence" of Volume 2). Last, diagnosis depends very directly on the means available for observing the system and research in diagnosis has direct links with systems design and observability and also their repairability if one is interested, as it is most often the case currently, in the design of autonomous and embedded systems, as well with their hardware features as their software ones.

2 Logical Framework for Diagnosis

The formalization of the theory of diagnosis at the end of the eighties has been firstly introduced separately regarding consistency-based diagnosis and regarding abductive diagnosis. In the first case, one requires only for a diagnosis, i.e., an assignment of behavioral modes – normal or abnormal – to each component of the system, to be consistent with the system model and the observations. In the second case, one requires additionally for a diagnosis to "explain", jointly with the system

420043_1_En_21_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:15/12/2018 Pages: 35 Layout: T1-Standard

model, all or some of the observations. Initially, this second case was most often handled in the framework of "naturally abductive" models such as causal graphs or Bayesian models and called on concepts of set covering. It is only a bit later that both approaches converged, the logical framework allowing the whole spectrum from simple consistency to whole abductive to be expressed.

2.1 Consistency-Based Logical Approach

The theory of consistency-based diagnosis was expressed for the first time in a logical framework, which will no longer vary afterwards, in Reiter (1987). This framework claims to be valid for any system structurally described in terms of components, the model of the system being assumed to be given by a first-order theory. One assumes likewise to have available a (sound and complete) first-order solver for checking inconsistency, which, in its whole generality, can be only a semi-algorithm as first-order theory is undecidable. The theory developed is completely independent from the choice of this solver, that we can suppose adapted to such and such actual systems modeling formalism according to their characteristics (but the practical tasks of aid to modeling and to inference algorithms specification are not tackled in this framework). On the other hand, the expression and the computation of diagnoses themselves from the results of the solver come under propositional logic, as the target vocabulary – components normality or abnormality – is propositional.

Definition 1 A *system* is a pair (*SD*, *COMPS*) where *SD*, the *system description*, is a finite set of first-order sentences (with equality) and *COMPS*, the *system components*, is a finite set of constants.

An *observations set OBS* is a finite set of first-order sentences (with equality).

An *observed system* is a triple (*SD*, *COMPS*, *OBS*) where (*SD*, *COMPS*) is a system and *OBS* an observations set.

The elements of *COMPS*, which are the subjects of the diagnosis, appear in *SD* and possibly in *OBS*. The behavioral mode or diagnostic status of each component is represented by a distinguished unary predicate *AB*(.), historically borrowed from the circonscription theory (McCarthy 1986), which is interpreted as signifying *abnormal*. The assumptions about components modes, which determine their behaviors, are thus made explicit in *SD* (nothing forbids *AB*(.) to appear also in *OBS*, but in practice it is always possible to transfer such an occurrence into *SD*). Typically, *SD* formulas code from one side the behavioral models of the generic components (library reusable for any system using the same components), in the form:

```
COMPONENT_TYPE(x) ∧ ¬AB(x) ⇒ Correct_model(x)
    /* correct functioning mode */
COMPONENT_TYPE(x) ∧ AB(x) ⇒
  Fault_model_1(x) ∨ … ∨ Fault_model_n(x) ∨ U(x)
```

```
        /* Fault modes */
   ¬Correct_model(x) ∨ ¬Fault_model_1(x), ...,
   ¬Fault_model_1(x) ∨ ¬Fault_model_2(x), ...
        /* exclusion in twos of the different behaviors*/
```

and from the other side the structural description of the system into its components in the form of ground formulas:

```
   INVERTER(C1), OR_GATE(C2), =(output(C1), input1(C2))
   RESISTOR(C3), =(resistance(C3),150).
```

`Correct_model(x)` is a formula that expresses the normal behavior of component `x`, while `Fault_model_i(x)` is a formula that expresses the behavior of component `x` for the fault mode `i`. The predicate `U(x)` is added to represent the unknown fault mode, thus not accompanied by any model, in order to express that the knowledge of the fault modes cannot claim in general to be exhaustive. It is important to notice that, as the theoretical framework does not assume anything about the nature of the formulas in *SD*, nothing requires modeling faults and one can be satisfied with the only correct functioning models. By the way, it is this idea that prevailed at the origin of model-based diagnosis: show that, unlike all the previous approaches (in particular expert systems) based on the knowledge of the faults and their effects, it was possible to do diagnosis without any prior knowledge of faults and symptoms.

As for the formulas in *OBS*, they describe measurements and are in general ground (but this is not mandatory), for example: `=(port2(C3),2.63)`.

A diagnosis is a mode assignment, normal or abnormal, to each component, which is consistent with both the system description and the observations. According to the context, a diagnosis will be identified either to a subset Δ of components (those that are abnormal) or to a conjunction $D(\Delta)$ of *AB*-literals, where the correspondence between $\Delta \subseteq COMPS$ and $D(\Delta)$ is defined by:

$$D(\Delta) = (\wedge AB(C) | C \in \Delta) \wedge (\wedge \neg AB(C) | C \in COMPS \setminus \Delta).$$

Definition 2 A *diagnosis* for $(SD, COMPS, OBS)$ is a $D(\Delta)$ with $\Delta \subseteq COMPS$ such that: $SD \cup OBS \cup \{D(\Delta)\} \not\models \bot$.

As there are potentially $2^{|COMPS|}$ possible diagnoses, one is often led to apply a parsimony principle and to be interested only in those diagnoses which are *minimal* for set inclusion (the subset of minimal size diagnoses may also be considered, but it is not in general the relevant concept).

Definition 3 A *minimal diagnosis* is a diagnosis $D(\Delta)$ such that $\forall \Delta' \subset \Delta$, $D(\Delta')$ is not a diagnosis.

Remark 1 A diagnosis for $(SD, COMPS, OBS)$ exists if and only if $SD \cup OBS$ is satisfiable, which will be always assumed in the following (otherwise, it means the model has to be revised). $\emptyset$ (i.e., $(\wedge\neg AB(C)|C \in COMPS)$) is a diagnosis (and the only minimal diagnosis) if and only if the observations are consistent with the correct functioning of all the components. Therefore fault detection occurs when $\emptyset$ is no more a diagnosis.

In order to locate the fault(s) after detection, it is natural to be interested in subsets of components – the minimal ones for set inclusion if possible – whose correct modes are by themselves (independently of the modes of the other components) inconsistent with the system model and the observations.

Definition 4 A *conflict set* for $(SD, COMPS, OBS)$ is a set $\mathscr{C} \subseteq COMPS$ such that $SD \cup OBS \cup \{\neg AB(C)|C \in \mathscr{C}\} \models \bot$. A *minimal* conflict set is a conflict set $\mathscr{C}$ such that $\forall \mathscr{C}' \subset \mathscr{C}, \mathscr{C}'$ is not a conflict set.

Each conflict set contains thus at least one abnormal component. Consequently a diagnosis Δ must have a nonempty intersection with each conflict set (one can restrain oneself to minimal ones).

Definition 5 Let $\mathscr{K}$ be a sets collection. A *hitting set* for $\mathscr{K}$ is a set $\mathscr{I} \subseteq \cup_{\mathscr{E} \in \mathscr{K}} \mathscr{E}$ such that $\forall \mathscr{E} \in \mathscr{K}, \mathscr{I} \cap \mathscr{E} \neq \varnothing$. A *minimal* hitting set is a hitting set $\mathscr{I}$ such that $\forall \mathscr{I}' \subset \mathscr{I}, \mathscr{I}'$ is not a hitting set.

Theorem 1 (Characterization of minimal diagnoses) *$\Delta \subseteq COMPS$ is a minimal diagnosis for $(SD, COMPS, OBS)$ if and only if Δ is a minimal hitting set for the collection $\mathscr{K}$ of minimal conflict sets for $(SD, COMPS, OBS)$.*

Theorem 1 provides an operational method for computing minimal diagnoses: one begins by computing all minimal conflict sets, then one computes the minimal hitting sets of the collection obtained in this way. An algorithm has been proposed by Reiter (1987) and corrected by Greiner et al. (1989), based on the construction and pruning of an acyclic direct graph (whose nodes are elements of $\mathscr{K}$ and labels of paths from the root to the leaves are the minimal hitting sets). As for the computation of all minimal conflict sets that involves an unsatisfiability test, it is in all generality a problem which is only semi-decidable; in practice, for real systems models, one deals with decidable fragments but the complexity class is in general NP-hard. An obvious but very inefficient algorithm would be to generate potential conflict sets candidates by a breadth first search of the lattice of subsets of $COMPS$, beginning by $COMPS$ (detecting a fault boils down to show that $COMPS$ is a conflict set and thus that $\emptyset$ is not a diagnosis), and continue by exploring the subsets of a set each time it has been proved to be a conflict set. This algorithm is improved by coupling conflict sets generation and minimal hitting sets computation: the call to the unsatisfiability checking solver is done at each node of the graph being developed by passing it as argument the conflict set candidate made up of the components that do not appear in the label of the path from the root to the node in question. One takes also advantage of the fact that the solvers (e.g., the resolution-based refutation method) may return, in case of unsatisfiability, the support of a refutation in the form of a conflict set that is in general strictly included into the conflict set passed as argument, which is used to label the node in question.

Actually, the most popular diagnostic architecture adopted by the majority of real implementations is the GDE (*General Diagnostic Engine*), introduced in De Kleer and Williams (1987) (simultaneously and independently from Reiter 1987). It rests on the coupling of a problem solver and an ATMS (*Assumption-based Truth Maintenance System*). Generally, the solver is based on constraints propagation: it propagates the values provided by *OBS* through the constraints expressing the system model *SD* (such a representation in the form of constraints, in particular equations from physics, is closer from models found in engineering than a first-order logic representation); that way it computes the output values of a component from its behavioral model equations and its input values. In this case the justifications transmitted to the ATMS are Horn clauses and the ATMS handles assumptions (namely the modes $AB(C)$ or $\neg AB(C)$ of each component) management by computing the labels (disjunctions of environments, where each environment is a conjunction of assumptions), supports of each statement inferred by the solver, in particular the *nogoods*, those environments that are the supports of $\perp$, i.e., the inconsistent assumptions sets. The framework of De Kleer and Williams (1987) is limited to the exclusive use of correct functioning modes: in the absence of faults modes, the assumptions are thus all of the type $\neg AB(C)$, which can be simply encoded by the propositional symbol C. In this framework and with this representation of assumptions, one obtains thus an equivalence between *nogoods* and conflict sets.

Property 1 *If only behaviors expressing necessary conditions of correct functioning are modeled in SD and the assumptions $\neg AB(C)$ are coded by the symbols C, then the minimal nogoods computed by an ATMS are exactly the minimal conflict sets.*

Moreover, in the absence of faults modes, one observes that changing, inside a minimal diagnosis Δ, the status $\neg AB(C)$ of a component C in $COMPS \setminus \Delta$ into $AB(C)$ cannot create any inconsistency, as no inference can be done from $AB(C)$. One obtains thus in this case a complete characterization of the set of diagnoses from the set of minimal diagnoses.

Property 2 *If only behaviors expressing necessary conditions of correct functioning are modeled in SD, then any superset of a diagnosis is a diagnosis. The diagnoses are thus exactly all supersets of the minimal diagnoses.*

In general, propagation is not a complete algorithm and one has to resort to more general constraints solvers, which lead to justifications that are no longer necessarily Horn clauses. In this case, and also for the explicit handling of the negation in the assumptions if faults modes are considered, an ATMS is no more sufficient and one has to use a CMS (*Clause Management System*) and to adapt the computation of the hitting sets.

To go further in the characterization of the set of diagnoses in the presence of faults modes, the concept of conflict set has to be generalized. For this, it is beneficial to move from a set representation of a conflict to a logical representation in the form of a clause, more precisely a *positive AB-clause* (disjonction of positive *AB-literals*).

Remark 2 A conflict set for $(SD, COMPS, OBS)$ identifies with a positive AB-clause $\vee_{C \in COMPS} AB(C)$ entailed by $SD \cup OBS$:

$$SD \cup OBS \models \vee_{C \in COMPS} AB(C).$$

Hence the immediate generalization:

Definition 6 A *conflict* for $(SD, COMPS, OBS)$ is an AB-clause entailed by $SD \cup OBS$, i.e., an AB-clause which is an *implicate* of $SD \cup OBS$. A *positive conflict* is a conflict whose all literals are positive. A *minimal conflict* is a *prime implicate*, i.e., a conflict whose no proper sub-clause is a conflict.

With this definition, the (minimal) conflict sets identify with the (minimal) positive conflicts. Thus the (minimal) hitting sets for the collection of minimal conflict sets identify with the (*prime*) *implicants* of the collection of minimal positive conflicts: one just has to identify the hitting set Δ with the *AB-conjunction* $\wedge_{C \in \Delta} AB(C)$. Moving from the set representation to the logical representation theorem 1 rephrases thus as:

Theorem 2 *$D(\Delta)$ is a minimal diagnosis for $(SD, COMPS, OBS)$ if and only if $\wedge_{C \in \Delta} AB(C)$ is a prime implicant of the collection of positive minimal conflicts for $(SD, COMPS, OBS)$.*

It is important to notice that as and when new observations appear, i.e., the set OBS is growing, the collection of positive conflicts increases as well and as a result some prime implicants do not remain any more in general. That is to say that some minimal diagnoses disappear and are replaced by other ones (involving more abnormal components). This means that the diagnostic process is *non-monotonic* as a function of the observations. This non-monotony is essential and actually it exists a close relationship between the diagnosis theory and the *default logic* (see chapter "Knowledge Representation: Modalities, Conditionals, and Nonmonotonic Reasoning" of this volume): one expresses that the components are correct in the form of (normal) defaults and one obtains a bijection between minimal diagnoses and extensions of the default theory built in this way.

Property 3 *Let $(SD, COMPS, OBS)$ be an observed system. Let DT be the following default theory: $DT = (\{: \neg AB(C)/\neg AB(C) | C \in COMPS\}, SD \cup OBS)$. Then E is an extension of DT if and only if $E = \{\pi | SD \cup OBS \cup D(\Delta) \models \pi\}$ where $D(\Delta)$ is a minimal diagnosis for $(SD, COMPS, OBS)$.*

The logical generalization of the concept of conflict allows one to characterize the set of all diagnoses, and not only of minimal diagnoses. One begins by defining a compact representation of the diagnoses, by considering the partial modes assignments to part of the components, such that all their extensions (by the modes normal or abnormal indifferently) to the rest of the components are diagnoses.

Definition 7 A *partial diagnosis* for $(SD, COMPS, OBS)$ is a satisfiable conjunction P of AB-literals such that, for any satisfiable conjunction P' of AB-literals containing P as a sub-conjunction, $SD \cup OBS \cup \{P'\} \not\models \perp$. A *kernel diagnosis* is a minimal partial diagnosis, i.e., none of its proper sub-conjunctions is a partial diagnosis.

With this definition, the kernel diagnoses provide a compact representation of all the diagnoses, these ones being exactly the total extensions of the kernel diagnoses.

Property 4 (Characterization of the diagnoses) *$D(\Delta)$ is a diagnosis if and only if it exists a sub-conjunction of $D(\Delta)$ which is a kernel diagnosis.*

Theorem 2 that characterizes the minimal diagnoses in terms of the positive conflicts is generalized as a characterization of the kernel diagnoses (and thus of all the diagnoses) in terms of the conflicts.

Theorem 3 (Characterization of the partial and kernel diagnoses) *The partial diagnoses (resp. kernel diagnoses) for $(SD, COMPS, OBS)$ are the implicants (resp. prime implicants) of the collection of minimal conflicts for $(SD, COMPS, OBS)$.*

Note that this theorem shows that the collection (in the disjunctive sense) of the kernel diagnoses, as a disjunctive normal form, is analogous to the collection (in the conjunctive sense) of the minimal conflicts, as a conjunctive normal form.

A sufficient condition guaranteeing that any superset of a diagnosis is a diagnosis has been given by the Property 2. Theorems 1 and 3 allow one to clarify the relationship between this property of closure of the diagnoses collection by the superset operation, and thus the complete characterization of diagnoses in terms of minimal diagnoses, and the nature of the conflicts.

Property 5 *There is a one-to-one correspondence between the kernel diagnoses and the minimal diagnoses (by extending any kernel diagnosis by the normal mode of all the components that it does not contain) if and only if all minimal conflicts are positive. More precisely, the two following statements are equivalent:*

1. *any superset of a minimal diagnosis is a diagnosis, i.e., if $D(\Delta)$ is a minimal diagnosis then $\forall \Delta$' such that $\Delta \subseteq \Delta' \subseteq COMPS$, $D(\Delta')$ is a diagnosis;*
2. *all minimal conflicts for $(SD, COMPS, OBS)$ are positive.*

Unfortunately one does not know an equivalent of the second statement of this property in terms of a syntactic characterization of $SD \cup OBS$. Only sufficient conditions guaranteeing the positivity of the minimal conflicts do exist, in the form of restrictions on $SD \cup OBS$. The most obvious one is to impose that any occurrence of an AB-literal in $SD \cup OBS$, put in conjunctive normal form, be positive. It is satisfied as soon as only the correct behavior of components is modeled, in the form of necessary conditions, which is the assumption of the Property 2.

Let add that in practice one limits oneself to compute the *preferred* diagnoses, according to a given criterion. It can be for example a *probabilistic* criterion if *prior* probabilities of the components behavioral modes are available. Diagnoses can

420043_1_En_21_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:15/12/2018 Pages: 35 Layout: T1-Standard

thus be generated in decreasing probability rank by using Bayes rule for evaluating conditional probabilities after each observation. One can use quantitative probabilities but also content oneself with relative orders of magnitude between probabilities. It can be also an *explanatory* criterion (see Sect. 2.2). Most of the time the preferred diagnoses are minimal and the selection according to the chosen preference criterion is thus done among minimal diagnoses, even for a model for which it is known that minimal diagnoses are not enough to characterize all diagnoses.

2.2 Abductive Approach

Graphs Based Approach

The very first approaches for diagnosis relied on causal models (see chapter "A Glance at Causality Theories for Artificial Intelligence" of this volume) representing in the form of arcs the causal relationships between the faults situations (D, for defects) that could affect the system and their effects, in particular their observable ones (M, for manifestations). Among these works, one can quote those from Reggia et al. (1983), Peng and Reggia (1990) which propose to use the covering sets theory to characterize the diagnoses. Arcs and nodes are associated to conditional probabilities and a plausibility measure is computed to rank the diagnoses.

Abductive Logical Approach

A limitation of the diagnosis approaches that are exclusively abductive is that they suppose *a priori* the "completeness" of the causal model, which has to describe all the faults and all the manifestations of these faults. An attempt to overcome this limitation is to take into account uncertain causal relationships by distinguishing strong causal link and weak causal link. Another one is, after having analyzed the differences between abductive and consistency-based approaches (Poole 1989), to try to reconcile them (Console and Torasso 1990). The idea is to distinguish among the observations those that the model has to explain (for example, the abnormal observations) from those whose only the consistency with the model is required (for example the exogenous or normal observations). It is examined in the papers (Console and Torasso 1991; Ten Teije and Van Harmelen 1994) which propose a synthesis of the various definitions that may result from it. This can be expressed in the same logical framework than previously by a logical diagnosis theory extending consistency-based diagnosis by abductive diagnosis. Similarly to Sect. 2.1, the following definitions, properties and theorems are obtained.

Definition 8 Let $(SD, COMPS, OBS)$ be an observed system and $OBS = I \cup O$ a partition of OBS, where O are those observations one wants to explain. An *abductive diagnosis* for $(SD, COMPS, I \cup O)$ is a $D(\Delta)$ with $\Delta \subseteq COMPS$ such that: $SD \cup I \cup \{D(\Delta)\} \not\models \bot$ and $SD \cup I \cup \{D(\Delta)\} \models O$.

420043_1_En_21_Chapter ✓ TYPESET DISK LE ✓ CP Disp.:15/12/2018 Pages: 35 Layout: T1-Standard

Definition 9 A *partial abductive diagnosis* for $(SD, COMPS, I \cup O)$ is a satisfiable conjunction P of AB-literals such that, for any satisfiable conjunction P' of AB-literals containing P as a sub-conjunction, $SD \cup I \cup \{P'\} \not\models \bot$ and $SD \cup I \cup \{P'\} \models O$. A *kernel abductive diagnosis* is a minimal partial abductive diagnosis, i.e., such that none of its proper sub-disjunctions is a partial abductive diagnosis.

Property 6 (Characterization of the abductive diagnoses) $D(\Delta)$ *is an abductive diagnosis if and only if it exists a sub-conjunction of* $D(\Delta)$ *which is a kernel abductive diagnosis.*

Theorem 4 (Characterization of the kernel abductive diagnoses) *Assume that SD, I and O are finite sets of formulas (each one being thus represented by a unique formula resulting from the conjunction of its elements). The kernel abductive diagnoses for* $(SD, COMPS, I \cup O)$ *are the prime implicants of* $\Pi \wedge \{(SD \wedge I) \Rightarrow O\}$, *where* Π *is the conjunction of the minimal conflicts for* $(SD, COMPS, I \cup O)$.

Notice that the logical concept of observations entailment used by the abductive diagnosis is unsuitable as soon as the observations are more precise than the predictions made from the models: one has in this case to resort to an abstraction of the observations (Cordier 1998; Besnard and Cordier 1994), represented by an observations lattice, and extend the definition of abductive diagnosis to that of explanatory diagnosis (explaining at best the observations).

2.3 Extensions

After having presented the formal framework of logical diagnosis, we quote rapidly below the issues that gave rise to later works.

When the number of diagnosis candidates is too large, it is important to use preference criteria to rank them. It is thus possible to generate the most probable diagnoses, from the *prior* faults probabilities (possibly qualitative) and use of Bayes rule (De Kleer 1992, 2006). One may also turn towards the sequential diagnosis, which consists in taking advantage of a succession of observations for reducing gradually the number of diagnoses. Some works had for purpose the choice of the best (in the sense of information theory, i.e., minimizing an entropy function) next observation in the framework of the sequential diagnosis. This issue meets the one of active testing (Feldman et al. 2009; Siddiqi and Huang 2010).

The diagnosis definitions and particularly the preferences (such as the probabilities) used to rank diagnoses are based in general on the assumption of faults independence. Some works are interested in the case of dependent faults such as cascading faults. A category of faults particularly difficult to diagnose is made up of faults affecting the structure (connectivity) of the system. Appear in this category the shortcuts between connections of a printed circuit board that result in hidden interactions (because not taken into account *a priori* in the model).

Rather early, when the application of the theory to real cases has been undertaken, arose the problem of handling uncertainty, both at the level of the model and at the level of the observations. It is especially important as the theory of consistency-based diagnosis only detects and makes explicit the causes of an inconsistency between the model of the system and the real system: inferring from that a malfunction of the system rests thus entirely on the correction of the model. Uncertainty is generally handled by resorting to an abstraction (Torta and Torasso 2003; Chittaro and Ranon 2004), or by qualitative models (that come under another important field in AI, the qualitative reasoning (see chapter "Qualitative Reasoning" of this volume)), or by expressing the values of the model parameters and of the observations by numerical intervals. According to the case, qualitative simulation or interval-based CSP are used as solvers (Dague et al. 1990).

Two research issues that emerged only at a later stage after the seminal works in the domain and are among the most active presently are diagnosability analysis and decentralized diagnostic architectures. The first, diagnosability, appeared around twenty years ago, arises from the assessment that the problem of designing and deploying a diagnostic architecture for a system must be tackled in advance at the very moment of the system design and not once the system has been produced and choices critical for the diagnosis, such as the number and the location of the sensors and thus the observation capacity of the system, have been fixed. For a given set of anticipated faults modeled in addition of the correct functioning of the system and a given set of observable quantities or events, the diagnosability analysis of the model answers the question to know if any occurrence of one of the faults will be always unambiguously identifiable in a finite time thanks to the observations only. Research in the field focused mainly on discrete event systems, modeled by transitions systems such as automata or Petri nets (see Sect. 3.6).

The second, more recent, concerns the diagnostic architectures either decentralized (local diagnosers communicating with a diagnostic supervisor in charge of providing the global diagnosis) or distributed (local diagnosers communicating between them for finding the global diagnosis), essential in particular for diagnosing systems that are by nature distributed (peer-to-peer networks, composite web services, etc.) but also systems made up of proprietary subsystems whose models are private for confidentiality reasons. Distribution may be related to the model, the observations, the algorithms, the software and hardware diagnostic architecture. Such architectures are presented in the case of discrete event systems in the Sect. 3.5.

Among the important problems, one can quote the preventive diagnosis, which consists in being able to detect a problem to come, before it occurs. This issue received attention later, probably because of the difficulty to get predictive models (such as wear models). Approaches different from model-based ones will have probably to be used in this case.

The issue of a tight coupling between diagnosis and repair or reconfiguration, critical in particular for autonomous systems, has been studied by using planning techniques (Sun and Weld 1993; Nejdl and Bachmayer 1993; Friedrich et al. 1994).

A last, important and difficult, problem is taking time into account. It is presented in the Sect. 3.1 and illustrated by the discrete event systems in the Sect. 3.2.

3 Diagnosis of Discrete Event Systems

3.1 Temporal Representation and Diagnosis

The previous section presents a theory that does not handle the representation of time and temporal reasoning. From this theory some extensions have been proposed that deal with several dimensions about time. Brusoni et al. propose in their paper *A spectrum of definitions for temporal model-based diagnosis* (Brusoni et al. 1998) a classification of these different extensions which take into account situations and successive observations as *time-varying contexts*; the system can also evolve between the production of two sets of observations (it is a *time-varying behavior*); faults can also produce observable effects after a given finite duration that can be represented as causal graphs (*temporal behavior*). Most of the time, these extensions can be represented by adding a time variable in the *SD* formulas associated with time constraints. Time is therefore reified. In practice, given a representation of the problem, it is necessary to look for compatible solvers that can manage inference and consistency tests by dealing with the selected representation of time (continuous, discrete or even both in hybrid systems).

Time variations in physical systems that are only due to system inputs do not add any new difficulty as this case can be interpreted as a discretized sequence of statical diagnosis problems. However, most of the systems are actually dynamical, they have internal states that memorize the past so that the behavior of the system not only depends on its current inputs but also on its current state. Time can be represented in a discretized way as a sequence of instantaneous events, in this case, the system is modeled as a discrete event system (see Sect. 3.2). Time can also be seen as a continuous variable that is described in differential equations, typically studied by the control theory community (FDI, see Sect. 4): AI and FDI methods have actually been compared (see Sect. 4). Based on the time granularity that is chosen in a model, continuous time can be symbolically abstracted as a set of instants that can be partially ordered, as time intervals, or as sequences of dates. In this last case, if the space of physical quantities is discrete, a concise representation of the temporal behavior can be done as a set of episodes, otherwise sequence of numerical intervals can be used. Some ATMS extensions are proposed to efficiently deal with these time data structures. It is possible to use the generic diagnosis theory that is described above by using an explicit variable that encodes time. However, the complexity of the model to acquire and the complexity of the inference and consistency test algorithms drastically increase. This theoretical framework can still be applied as long as the faults within the system are permanent (always present). If faults occur at the supervision time and if their effect is permanent after their occurrence, there is no fundamental changes as the evolution of the conflict sets is still monotonic. Dealing with intermittent faults is more difficult and is possible only if the evolution of such intermittent faults is slower than the evolution of the system itself and the speed of observation acquisition.

Most of the contributions, even the ones dealing with time, aim at solving the diagnosis problem based on observation logs after the system has stopped: this is the *off-line diagnosis* problem. Then the AI and FDI communities independently started to develop some works about *on-line diagnosis*. The system is observed at operating time in order to react (repair, control) when a discrepancy with the expected behavior is detected and maintain an operating state that is as satisfactory as possible.

Two types of methods can be distinguished.

1. In *chronicle recognition* methods, the objective is to recognize, within the flow of observations, some observable patterns that caracterize faulty situations. A chronicle is a set of events associated with time constraints. Specialized algorithms perform on-line chronicle recognition so that a decision about how to react after a fault has been diagnosed can still be made at operating time (Dousson 1996; Carle et al. 2011).
2. The second type of methods, that is typically model-based, relies on the behavioral description of the system but has to deal with the on-line observation flow incrementally. Assuming that only one fault has occurred or is permanent within the supervision time is not realistic as the supervision time is long. Moreover it must be considered that some faults are repaired during the supervision.

In the next sub-section, we focus on the methods where the system is modeled as a discrete event system (DES). This type of models is particularly relevant when the underlying system reacts to events (reactive systems), such that the opening/closing of a valve, the reception of messages, the occurrence of a fault. This type of models can also be relevant even if the system is continuous but can be discretized as a DES (Lunze 1994). From the initial work from Sampath et al. (1996), a set of contributions are proposed about the diagnosis of discrete event systems in the AI community as well as in the FDI community.

3.2 Models of Discrete Event Systems

A DES is a dynamical system whose state can be described by state variables and the domain of each variable is discrete. The behavior of the DES is caracterized by the occurrence of discrete events that instantaneously modify the internal state of the DES. This representation is obviously well-suited to describe systems that are naturally discrete, such as communication networks that aim at receiving, sending messages, automated production line systems that produce objects step by step, etc. But this representation is also well-suited for systems that can be discretized, resulting for example from a qualitative reasoning method (Travé-Massuyès and Dague 2003).

To model DES, several formalisms from the language theory can be used such as the process algebra, Petri nets and automata. With the help of these formalisms, the behavioral language of the DES can be represented in a concise manner. In order to present and illustrate the diagnosis problem of DES, we use here the formalism of *transition system/automaton* (see chapter "Theoretical Computer Science: Computational Complexity" of Volume 3) which has been used in most of the seminal works of the field.

Definition 10 An *automaton* A is a 5-tuple $\langle Q, E, T, I, F\rangle$ such that

- Q is a finite set of states,
- E is a finite set of events,
- $T \subseteq Q \times E \times Q$ is a finite set of transitions $\langle q, e, q'\rangle$,
- $I \subseteq Q$ is a set of initial states,
- $F \subseteq Q$ is a set of final states.

The event e over the transition $t = \langle q, e, q'\rangle$ triggers the transition. The language $L(A) \subseteq E^\star$ generated by the automaton A is the set of event sequences from E which can be associated with a transition path in A from an initial state q_0 of I to a final state of F, such a path is also called a *trajectory*.

Definition 11 A *trajectory* of an automaton $A = \langle Q, E, T, I, F\rangle$ is a sequence of transitions $traj = q_0 \stackrel{e_1}{\to} \dots \stackrel{e_m}{\to} q_m$ such that: $q_0 \in I$, $q_m \in F$, and $\forall i \in \{1, \dots, m\}$, $\langle q_{i-1}, e_i, q_i\rangle \in T$. A trajectory can also be denoted as $\langle (q_0, \dots, q_m), (e_1, \dots, e_m)\rangle$.

The set of possible behaviors of a system is represented as an automaton *SD*, each behavior being caracterized as a trajectory in *SD*.

Definition 12 The *model of the system* is an automaton

$$SD = \langle Q^{SD}, E^{SD}, T^{SD}, I^{SD}, F^{SD}\rangle.$$

As any trajectory $q_0 \stackrel{e_1}{\to} \dots \stackrel{e_m}{\to} q_m$ of the system depends on a previous trajectory of the system $q_0 \stackrel{e_1}{\to} \dots \stackrel{e_{m-1}}{\to} q_{m-1}$, the *SD* automaton can then be such that $F^{SD} = Q^{SD}$ (any state is final). In other words, the language $L(SD)$ is prefix-closed.

In general, a DES can be modeled in a modular way as a set of n components $COMPS = \{C_1, \dots, C_n\}$ that define the *structural model* of the supervised system. Each component C_i is modeled as an automaton $SD_i = (Q_i, E_i, T_i, I_i, F_i)$. The model of the system is obtained by applying a *synchronized product* on the automata $(SD_i)_{i=\{1,\dots,n\}}$. The product relies on a set of *synchronisation relations Sync* that are generally a set of constraints $e_i = e_j$ that model the fact that the event e_i of C_i and the event e_j of C_j must always occur at the same time. The global model *SD* is obtained by computing the subset of trajectories from the Cartesian product $\prod_{i=1}^{n} SD_i$ that is restricted to the trajectories when all the constraints of *Sync* are satisfied. This synchronized product is denoted $\otimes_{Sync}$ or simply $\otimes$ when the synchronisation constraints are defined without ambiguity. From this, it follows:

$$SD = SD_1 \otimes \dots \otimes SD_n.$$

3.3 Faults, Observations and Diagnosis of DES

The automaton *SD* that represents the system, actually models its normal and abnormal behaviors, and especially the behaviors of interest in the monitoring task. The abnormal behaviors are modeled by labeling transitions with *fault events* $e_f \in F \subseteq E^{SD}$ that represent the fact that the system starts to be faulty.

Any diagnosis reasoning requires the observation of the system. In the context of the DES, observations are events, usually resulting from the generation of a piece of information from sensors. In a DES, there are observable events $E^{SD}_{OBS} \subseteq E^{SD}$ and non-observable events $E^{SD}_{\neg OBS} \subseteq E^{SD}$. Among the non-observable events, there are the fault events. Any trajectory τ of the system is then associated with its *observable trace* $\sigma(\tau)$ that is defined as the sequence of observable events that is produced when τ is indeed the trajectory realized by the system (projection of τ on the observable events E^{SD}_{OBS}).

If it is assumed that the observations of the system are perfectly known (no uncertainty about the observed event types and the observed dates), the observation of the system is then defined as *a sequence of observable events*.

Definition 13 The *observation* of the system, denoted *OBS*, is the sequence of observable events that is produced by the system within the time frame of the diagnosis reasoning.

The diagnosis task consists in comparing the effective observation of the system with the prediction of the model as the possible set of observable traces, and then to determine the set of non-observable events (especially the fault events) that explain the current state of the system (Cordier and Thiébaux 1994).

Definition 14 A *diagnosis problem* is described as a 3-uple (SD, OBS, F) where *SD* is the model of the system, *OBS* is the observation of the system et F is a set of fault events.

In order to determine the faults, it is firstly necessary to search for the set of system's trajectories in the model *SD* whose observable trace matches *OBS* exactly.

Definition 15 (*Trajectory Diagnosis*) A diagnosis Δ for the problem (SD, OBS, F) is a trajectory of *SD* whose observable trace $\sigma(\Delta)$ is exactly *OBS*.

With this definition, the diagnosis problem does not depend on faults (it can be defined as a couple (SD, OBS)). However, the diagnosis can also be defined in a more concise way as a set of faults. This second definition is closely related to the one for statical systems.

Definition 16 (*Fault Diagnosis*) A diagnosis Δ of the problem (SD, OBS, F) is a set of faults $\Delta \subseteq F$ such that there exists a trajectory τ from *SD* that exactly contains the set of fault events Δ and its observable trace $\sigma(\tau)$ is exactly *OBS*.

It can be noticed that the set of trajectory diagnoses of a system can also be represented as an automaton, more precisely it is a sub-automaton of *SD*, each trajectory in it has an observable trace that is exactly *OBS*.

Fig. 1 Model of the system, o_b, o_c, r are the observable events

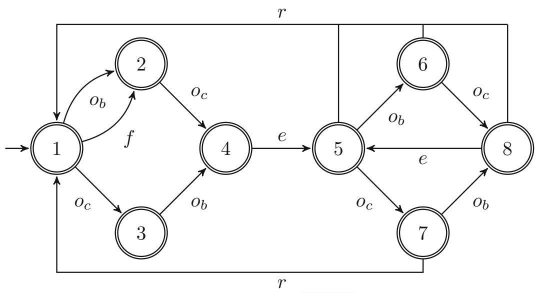
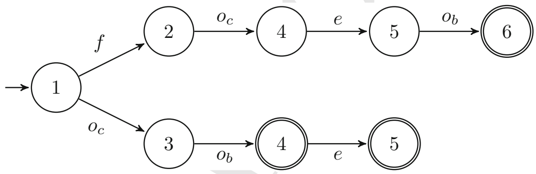


Fig. 2 Diagnoses of the system (Fig. 1) given the observed sequence $OBS_1 = (o_c, o_b)$



Example 1 Figure 1 illustrates a system with a set of observable events o_b, o_c and r. If the observed sequence is $OBS_1 = (o_c, o_b)$, the set of diagnoses are the one presented as an automaton in Fig. 2, the fault f is not certain and the possible states of the system are 4, 5, 6. If the observed sequence is $OBS_2 = (o_c, o_c)$, the unique diagnosis is presented in Fig. 3: the occurrence of the fault f is indeed certain and the unique possible state is 7.

An observation *OBS* consisting of a sequence of observed events can also be represented as an automaton with one initial state and one final state. The diagnosis can then be computed by performing a *synchronized product* $\otimes$ between the automaton *SD* and the one that describes *OBS*. The synchronization constraints *Sync* are applied on the observable events: an observable event o must occur in *SD* and in *OBS* in the same order. Representing *OBS* this way is interesting as it can be extended to represent uncertain observations. In this case the automaton *OBS* does not represent one sequence of observable events only but several possible sequences (Grastien et al. 2005). From this follows the next theorem:

Theorem 5 *The automaton* $SD \otimes OBS$ *describes the set of trajectory diagnoses from the problem* (SD, OBS).

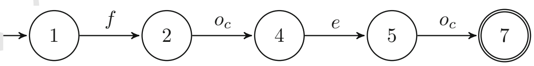


Fig. 3 Diagnosis of the system (Fig. 1) given the observed sequence $OBS_2 = (o_c, o_c)$

3.4 Diagnoser Approach and Other Centralized Approaches

One of the seminal works to compute diagnosis on DES is in Sampath et al. (1996) and is based on the computation of a *diagnoser* (see Fig. 4). A diagnoser is a deterministic automaton that describes the set of observable behaviors of the system in a similar way as an observer would do. It is built by ε-reduction from the automaton *SD* where ε represents any non-observable event of *SD*. A diagnoser transition is labeled with an observable event. A state of a diagnoser describes the set of states of *SD* that are reachable from its initial states and that are reachable by trajectories that produce the observable sequence. Associated with each state of *SD* the diagnoser state also records sets of fault events that have occurred on such trajectories. For a given sequence of observable events, the diagnoser state thus describes the set of possible reached states and the set of possible faults that have occurred before reaching one of these states.

The diagnoser is a finite state machine that results from the off-line compilation of the diagnosis problem and its use for on-line diagnosis is performed by a simple algorithm. Indeed, the on-line algorithm consists in triggering the observed events of *OBS* in sequence and the result of the algorithm is contained in the diagnoser state that is reached. The problem of this method is about the time/space complexity of the computation of the diagnoser. In Marchand and Rozé (2002), Schumann et al. (2004), other computation methods have been proposed to improve the efficiency on average of the diagnoser computation. These methods rely on binary decision diagrams (*BDD*).

Other methods use different formalisms to build an equivalent diagnoser such as communicating automata, Petri nets, process algebra (Rozé and Cordier 2002; Jiroveanu and Boel 2006; Console et al. 2002). Other works (Lamperti and Zanella 2003) propose specialized data structures and specific algorithms to solve the diagnosis problem. On the other hand, Grastien and Anbulagan (2013) propose the use

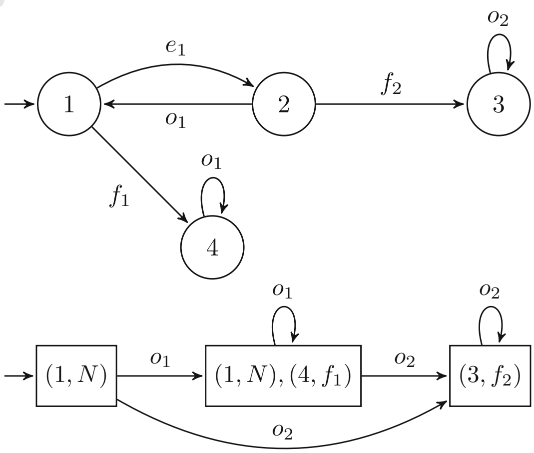


Fig. 4 The global model *SD* (with $E^{SD}_{OBS} = \{o_1, o_2\}$ and $F = \{f_1, f_2\}$) (Top) and its diagnoser (Down). Label *N* (normal) means the absence of any fault

of generic SAT techniques and translate the diagnosis problem into a succession of propositional formulas (CNF). It is also possible to use probabilistic models that can model the likelihood of transitions between states. One preference criteria is then to keep the transitions that are the most probable, this can be done for instance by applying the Viterbi algorithm such as in Aghasaryan et al. (1997).

Three extensions of the classical diagnosis problem have been mainly investigated. In the first one, the hypothesis that *OBS* is certain is removed (Lamperti and Zanella 2003; Grastien et al. 2005). It is, in this case, impossible to assert that there is a unique sequence of observations, either because the knowledge about the real order of the observed events is not perfect or because events can be lost or corrupted (noise). This might be due to imprecise or even faulty sensors, or the communication network between the sensors and the diagnoser. One solution then consists in representing the observations as an automaton that contains the set of possible observed trajectories. Then Theorem 5 can be used as in Grastien et al. (2005). The second extension of the problem is about on-line diagnosis that is well-suited for the on-line monitoring of dynamical systems such as communication networks. In this context, *OBS* is partly known (a prefix of *OBS* is known). On-line diagnosis then leads to incremental diagnosis that consists in updating the diagnosis from a previous diagnosis in a new time window when new observed events are available (Pencolé and Cordier 2005; Grastien et al. 2005). Incremental diagnosis has also be extended to deal with large scale systems where it is not possible to efficiently update the diagnosis with the flow of observations. In Su and Grastien (2013), the principle is to compute a diagnosis for a given time window independently from any other time window and Su et al. (2014) analyses the minimal amount of information to retain between time window to assert the diagnosis is correct along the time. Finally, a more recent extension is about the diagnosis of behavioral patterns (Jéron et al. 2006; Pencolé and Subias 2018). In the classical problem, the model represents faults as the occurrence of single events. With behavioural patterns, it is also possible to represent in the model a set of events that might not be considered independently as faulty but some specific ordering of their occurrence can still be abnormal (for instance, in traffic light systems, the sequence of *green, yellow, red* is normal while *green, red, yellow* is not).

3.5 Distributed and Decentralized Approaches

Most of the systems that are monitored and diagnosed have a large size so that a centralized method, as described in the previous sub-section, is not efficient enough. To illustrate this inefficiency, it can be noticed that the synchronized product over the components'models is in $O(2^n)$ where n is the number of components. This complexity makes a centralized approach impossible to implement on a realistic system. Based on the distributed nature of a system as a network of components, it is then possible to design decentralized or even distributed diagnosis methods that are more scalable. The model is then described as a set of components'models and a set of connections and the global model is not explicitly computed. Several formalisms have been proposed to model the system in a distributed way: automata where the connections are represented by shared events, communicating automata where the connections are represented by messages on input/output ports (Pencolé and Cordier 2005), Petri nets where interactions are modeled by shared transitions or places (Fabre et al. 2005; Jiroveanu and Boel 2006), process algebra (Console et al. 2002) where the synchronization is represented as a cooperation operator.

There exist several methods implementing the collaboration of several local diagnosers to solve a diagnosis problem. Several types of methods can be distinguished depending on the supervision architecture. In a so-called *coordinated* architecture, each diagnoser is in charge of observing local sites and determining a global diagnosis based on its observation sites. Then a coordinator analyzes the global diagnoses of each diagnoser and provides a unique and coordinated one (Debouk et al. 2002). In this type of architecture, local diagnosers must still know the global model of the system so such an architecture has a scalability issue. A second architecture where the local diagnosers do not need to know the global model is the *decentralized* architecture. As opposed to the coordinated architecture, local diagnosers only have a local knowledge about the system (a subset of components, also called a cluster). Local diagnosers perform diagnosis only over the components they know. The local diagnoses, once computed by the local diagnosers, are sent to a global diagnoser that is in charge of checking the global consistency of the local diagnoses (Pencolé and Cordier 2005; Lamperti and Zanella 2003; Grastien et al. 2005; Pencolé et al. 2018) by checking whether the local diagnosed trajectories are globally synchonizable. The last investigated architecture is the *distributed* architecture. The main difference with the decentralized architecture is that there is no global diagnoser. The result of the diagnosis is not global but only local. Each local diagnoser is in charge of handling the global consistency of its diagnosis by interacting with other local diagnosers (Fabre et al. 2005). The diagnosis algorithms then depend on the selected architecture. Computing a global diagnosis might be a necessity to decide about a global repair or a global reconfiguration of the system, in this case, coordinated or decentralized methods should be used. If the decision is local then a distributed architecture is sufficient.

In the case of distributed or decentralized architectures, the complexity of the algorithms mainly depends on checking the global consistency of the local diagnoses that depends on the number of involved components. To improve this global consistency checking, a BDD-based synchronization algorithm is proposed in Schumann et al. (2010). Another way to increase the average efficiency of the algorithms is to analyze off-line the structural model of the system (the topology) to precompile basic synchronization strategies that can be then applied on-line. For instance in KanJohn and Grastien (2008), this analysis is based on junction trees. In Pencolé et al. (2006), the analyzis consists in determining off-line clusters of components based on which a local diagnoser is always *accurate*: it is certain to always have a global consistent diagnosis without any synchronization with other components out of the given cluster (Fig. 5). AQ2

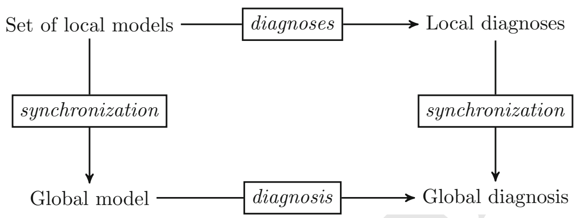


Fig. 5 Principle of a decentralized diagnosis

3.6 Diagnosability

The off-line analyses of DES properties related to the diagnosis problem is essential to implement efficient on-line diagnosis algorithms (such properties like diagnosis accuracy of clusters, topology properties as cited in the sub-section above). Among these properties, *diagnosability* is the most studied one (see Sect. 2.3).

Intuitively, in the context of DES, a system is diagnosable if, in case of an ambiguous diagnosis (faulty or not) at a given time, it is always sufficient to wait for a new finite set of observations to refine the diagnosis and prune the ambiguity and obtain a diagnosis that is certain. The first formal definitions of this property are proposed in Sampath et al. (1995). Several extensions have then be defined, by considering intermittent faults (Contant et al. 2004), or by extending faults to behavioral patterns (Jéron et al. 2006; Gougam et al. 2017; Ye and Dague 2017). A definition that unifies the one for continuous systems and the DES is proposed in Cordier et al. (2006). Checking the diagnosability is a complex problem. The first solution is presented in Sampath et al. (1995) and consists in checking in the diagnoser (see Fig. 4) whether there is no indeterminate cycles (cycles of states where the diagnosis is ambiguous). Then another solution that is polynomial in the number of states in the global model is based on the synchronization of the model with itself (some *twin models*) where the synchronizations are performed on the observable events only. It consists in checking in this product for infinite sequence of critical pairs (a sequence that represents a faulty sequence in one twin and a non-faulty sequence in the other one) (Jiang et al. 2001; Yoo and Lafortune 2002). Other algorithms improving the efficiency of this method can also be found in Cimatti et al. (2003), Schumann and Pencolé (2007).

One of the objectives of checking diagnosability is to provide a feedback to the design of the system, by essentially adding new sensors (Travé-Massuyès et al. 2001; Ribot et al. 2008), or by respecifying communication protocols between components of the system (Pencolé and Cordier 2005). Some extensions about the diagnosability of distributed systems are also proposed in Provan (2002), Pencolé (2004), Ye and Dague (2012). One method also extends the diagnosability problem to deal with uncertain observations (Su et al. 2016). Another extension is about *self-healability* that combines diagnosability and repairability. A system is said to be self-healing if it is able to perceive its own faults and, without any human intervention to perform necessary actions to recover. Self-healability can hold in a system even if the system is not fully diagnosable and not fully repairable. The required level of diagnosability is the one that can always make the repair decision certain. In Cordier et al. (2007), this level of diagnosability is based on a selection of *macrofaults* are diagnosable and repairable.

4 Bridge Between Model-Based Diagnosis Rooted in AI and in Automatic Control

In the field of DES, the AI community (known as DX community) and the Automatic Control community (known as FDI – *Fault Detection and Isolation*–community) have converged from the start on the same formalisms and jointly developed diagnosis methods. On the contrary, for continuous systems, these communities have worked in parallel for a long time, ignoring their respective results. Although there are common principles, each community has developed its own concepts and methods, guided by different modeling approaches, and relying on analytical models and linear algebra for the first and on logical formalisms for the latter. However, in the 2000s, under the impetus of the BRIDGE group "*Bridging AI and Control Engineering model based diagnosis approaches*" within the Network of Excellence MONET II and its French counterpart, the IMALAIA group "*Integration of Methods Combining Automatic Control and AI*" linked to GDR I3, the French Association for Artificial Intelligence AFIA, and GDR MACS, an increasing number of researchers from these two communities have sought to understand and integrate approaches of their respective fields to provide more effective diagnostic systems (Travé-Massuyès 2014).

First of all, we draw up a panorama of the approaches proposed by the FDI community, and then present a comparative analysis of the concepts and techniques used in the two communities in Sect. 4.2, followed by the works which integrate techniques of both communities in Sect. 4.3.

4.1 FDI Community and Approaches for Continuous Systems: Quick Panorama

Like the methods of the DX community (cf. Sect. 2), the fault detection and diagnosis methods of the FDI community are based on behavioral models that establish the constraints between the system inputs and outputs, i.e., the set of measured variables Z, as well as the internal states, i.e., the set of unknown variables X. The variables $z \in Z$ and the variables $x \in X$ are functions of time. These models are formulated either in the time domain (then known as *state space models*) or in the frequency domain (then known as *transfer functions* in the linear case).

420043_1_En_21_Chapter ✓ TYPESET ☐ DISK ☐ LE ✓ CP Disp.:15/12/2018 Pages: 35 Layout: T1-Standard

The books (Gertler 1998; Blanke et al. 2003, 2015; Dubuisson 2001) are very good reviews that include the references to original papers, to which the reader can refer.

The central concept of FDI methods is that of *residual* and one of the main problems is the *generation of residuals*. Consider the model of a system under the form of a set of differential and/or algebraic equations $SM(z, x)$ with variables z and x. $SM(z, x)$ is said to be *consistent with the observed trajectory* z, if there exists a trajectory of x such that the equations of $SM(z, x)$ are satisfied.

Definition 17 (*Residual generator for* $SM(z, x)$) A system that takes as input a subset of measured variables $\tilde{Z} \subseteq Z$ and generates as output a scalar r, is a residual generator for the model $SM(z, x)$ if for all z consistent with $SM(z, x)$, $\lim_{t\to\infty} r(t) = 0$.

When the measurements are consistent with the system model, the residuals tend to zero as t tends to infinity, otherwise some residuals may be different from zero. Evaluating the residuals and assigning them a Boolean value – 0 or non-0 – requires statistical tests that account for the statistical characteristics of noises (Dubuisson 2001; Gao et al. 2015). There are three main families of methods for generating residuals.

- The *methods based on testable relations* rely on unknown variables elimination. These methods generate residuals from relations inferred from the model which only involve measured variables and their derivatives. These relations are called *Analytical Redundancy Relations* (ARRs). For linear systems, the so-called *parity space* approach is used to eliminate unknown state variables and obtain ARRs by projection onto a particular space called the parity space (Chow and Willsky 1984). Extensions of this approach to nonlinear systems have been proposed (Staroswiecki and Comtet-Varga 2001). The structural approach (Armengol et al. 2009) allows one to obtain the just determined equation sets of a model from which ARRs can be inferred (Krysander et al. 2008).
- The *methods based on state estimation* are based on estimating unknown variables. They take the form of *observers* or optimized *filters*, such that the Kalman filter, and provide an estimation of the state of the system and its outputs. Numerous diagnosis solutions rely on state estimation, particularly for hybrid systems (cf. Sect. 4.3). In this case, the continuous state is augmented by a discrete state that corresponds to the operation mode (normal or faulty) of the system components.
- The *methods based on parameter estimation* focus on the value of the parameters which directly represent physical characteristics. Fault detection is performed by comparing the estimated value of the parameters to their nominal value. These methods are used for both linear and nonlinear systems.

Note that in the linear case, the equivalence between observers, parity space and parameter estimation has been established (Patton and Chen 1991).

4.2 Comparative Analysis and Concept Mapping for the Model-Based Logical Diagnosis Approach and the Analytical Redundancy Approach

The correspondences in terms of principles, concepts, and assumptions between the model-based diagnostic methods from Automatic Control and those from AI were showed by the French community, concretized by the IMALAIA group mentioned above. This work is recorded in the collective paper (Cordier et al. 2004). The comparative analysis is based on the comparison of the so-called structured residuals approach, or parity space approach (Chow and Willsky 1984), and the logical theory of diagnosis as proposed by Reiter (1987), Kleer et al. (1992) and presented in Sect. 2.

The parity space approach is based on the off-line computation of a set of ARRs from a model SM decomposed in a behavior model BM and an observation model OM. The equations of the model SM are constraints which can be associated with components but this information is not represented explicitly.

The ARRs define constraints for the observable variables O of the system, that is to say the input and output variables, and are obtained by techniques allowing to eliminate state variables that are unknown. Each ARR can be put in the form $r = 0$, where r is called *residual*.

Definition 18 (*ARR for $SM(z, x)$*) A relation of the form $r(z, \dot{z}, \ddot{z}, \dots) = 0$ is an ARR for the model $SM(z, x)$ if for all z consistent with $SM(z, x)$, the relation is satisfied.

If the behavior of the system satisfies the constraints of the model, then the residuals are zero because the ARRs are satisfied, otherwise some of them may be different from zeros and the corresponding ARRs are said violated. Each fault F_j has an associated theoretical signature $FS_j = [s_{1j}, s_{2j}, \dots, s_{nj}]$ given by the binary evaluation (0 or not 0) of each of the residuals. We can then define the *signature matrix* FS.

Definition 19 (*Signature Matrix*) Given a set of n ARRs, the signature matrix FS associated to a set of n_f faults $F = [F_1, F_2, \dots, F_{n_f}]$ is the matrix that crosses ARRs as rows and faults as columns, and whose columns are given by the theoretical signatures of the faults.

Diagnosis consists in the online comparison of the "observed signature", vector of the residuals evaluated with the observations, and the theoretical signatures of the n_f anticipated faults. In the logical theory of diagnosis, the description of the system is component oriented and rests on first order logic in its original version. This has been discussed in detail in Sect. 2.1.

A diagnosis for the system $(SD, COMPS, OBS)$ is a set $\Delta \subseteq COMPS$ such that the assumption that the components of Δ are the only ones to be faulty is consistent with the observations and the description of the system, that is $SD \cup OBS \cup \{AB(C) \mid C \in \Delta\} \cup \{\neg AB(C) \mid C \in COMPS \setminus \Delta\}$ is satisfiable.

Most FDI works do not explicitly use the concept of component given that the behavior model BM represents the global system. When models based on the concept of component are used, topological knowledge is implicitly represented by shared variables. Conversely, the DX approach explicitly represents the topology of the system and the behavior models of the components. The main difference is that the hypothesis of correct behavior of a component, which underlies its model, is represented explicitly by the predicate AB. If $\mathscr{F}$ is a formula representing the correct behavior of a component C, SM contains only $\mathscr{F}$ while SD contains the formula $\neg AB(C) \Rightarrow \mathscr{F}$.

To compare the approaches, the *system representation equivalence* (SRE) property resulting in the fact that SM is obtained from SD by substituting all the occurrences of the predicate $AB(.)$ by $\perp$ is considered true. It is also assumed that the same observation language OBS is used, constituted by a conjunction of equality relations that assign a value v to each observable variable. Finally, the faults relate to the same entities considered as components, without loss of generality. The comparison is based on a theoretical framework to precisely establish the correspondence between the different concepts. This framework is provided by the signature matrix FS, for which each row is associated with an ARR and each column with a component (under the assumption that the faults relate to components). It relies on the concept of *support* of an ARR:

Definition 20 (*ARR Support*) The *support* of an ARR ARR_i, noted $supp(ARR_i)$, is the set of components whose columns in the signature matrix FS have a non zero element on the ARR_i row.

In addition, the following two properties are added:

Property 7 ARR–d–completeness *A set E of ARRs is said to be* d-complete *if:*

- *E is finite;*
- *$\forall OBS$, if $SM \cup OBS \models \perp$, then $\exists ARR_i \in E$ such that $\{ARR_i\} \cup OBS \models \perp$.*

Property 8 (ARR–i–completeness) *A set E of ARRs is said to be* i-complete *if:*

- *E is finite;*
- *$\forall \mathscr{C}$, set of components such that $\mathscr{C} \subseteq COMPS$, and $\forall OBS$, if $SM(\mathscr{C}) \cup OBS \models \perp$, then $\exists ARR_i \in E$ such that $supp(ARR_i)$ is included in $\mathscr{C}$ and $\{ARR_i\} \cup OBS \models \perp$.*

We then obtain the following result:

Property 9 *Assuming the SRE property and that OBS is the set of observations for the system given by SM (or SD), then:*

1. *If ARR_i is violated by OBS, then $supp(ARR_i)$ is a conflict set;*
2. *Given E a set of ARRs:*
    - *If E is d-complete, and if there exists a conflict set for $(SD, COMPS, OBS)$, then there exists $ARR_i \in E$ violated by OBS;*

- *If E is i-complete, then given a conflict set* $\mathscr{C}$ *for* $(SD, COMPS, OBS)$*, there exists* $ARR_i \in E$ *violated by OBS such that* $supp(ARR_i)$ *is included in* $\mathscr{C}$.

The first result can be intuitively explained by the fact that inconsistencies between model and observations, appraised by the conflicts in the DX approach, are apprehended by ARRs violated by *OBS* in the FDI approach. In consequence, the support of an ARR can be defined as a *potential conflict*. This result echoes the notion of *possible conflict* proposed in Pulido and Gonzalez (2004). The second result provides existence and completeness results, the first referring to detectability and the second to isolability.

We then show below that in the presence of the same assumptions about the manifestation of faults (their observability), commonly called *exoneration assumptions*, and in particular the absence of ARR-exoneration, a result linking the diagnoses on both sides can be obtained.

Definition 21 (*ARR-exoneration*) Given *OBS*, any component in the support of an ARR satisfied by *OBS* is exonerated, i.e., considered as normal.

This assumption states that faults having no observable manifestation through a non-zero residual are exonerated.

Theorem 6 *Under the i-completeness assumption, the diagnoses obtained by the FDI approach in the case of no ARR-exoneration are identical to the (non empty) diagnoses obtained by the DX approach.*

Let us note that the assumptions generally adopted by the two communities are different, the FDI community implicitly adopting the ARR-exoneration assumption. In addition, the computation of fault signatures limits the number of anticipated faults. Conventionally, only single faults are considered. Conversely, in the logical diagnosis theory, no assumption is made *a priori* about the number of faults, even if preferences can be introduced to privilege minimal or highest probability diagnoses. This ensures logically correct results. It can also be noted that in the FDI approach, computation of ARRs and fault signatures is done offline and only a consistency test is required online. This can be advantageous if computational time constraints come into play. In the logical theory diagnosis, all the processing is done online, the advantage being that only the models are to be updated if the system undergoes changes. Note that the two approaches can be combined to take advantage of both. One can cite DX works which adopt the FDI idea of offline generation of the RRAs (Loiez and Taillibert 1997; Washio et al. 1999; Pulido and Gonzalez 2004). One can also cite the works, presented in more detail in the Sect. 4.3, which take advantage of explicitly representing the causal influences underlying the model of the system and those concerned with diagnosis of hybrid systems.

4.3 Approaches Taking Advantage of Techniques of Both Fields

Diagnosis Based on Influence Graphs/Causal Graphs

In the 1990s, the synergies between the Qualitative Reasoning community (Travé-Massuyès and Dague 2003; Weld and De Kleer 1989) (see also chapter "Qualitative Reasoning" of this volume) and the Model-Based Diagnosis community concretized in a set of works proposing to use *causal models* for diagnosis reasoning (see chapter "A Glance at Causality Theories for Artificial Intelligence" of this volume). Unlike causal graphs pointed in Sect. 2.2, influence graphs rely on a structure expressing the dependencies between variables in the model of the system explicitly, known as *influences* thus making it possible to provide explanations as to why normal or abnormal values of variables. This structure is commonly called a *causal graph*. Dependencies are obtained directly from expert knowledge (Gentil et al. 2004) or from causal ordering techniques (Travé-Massuyès et al. 2001; Pons et al. 2015) or also from *bond graph* models (Dague and Travé-Massuyès 2004; Chatti et al. 2014).

The very first works were limited to labeling the causal influences by the signs giving the direction of variation of the cause variable with respect to the effect variable, thus obtaining a *signed oriented graph* (Kramer and Palowitch 1987). Subsequently, the parametrization of influences was sophisticated as they were labeled by quantitative local models, such as those used by the FDI community.

By way of example, the principles of the causal fault detection and isolation method CaEn2 (Travé-Massuyès et al. 2001; Travé-Massuyès and Calderon-Espinoza 2007) are given below. Fault detection is an online process that assesses the consistency of sensor measures with respect to the behavioral model of the system. The detection of a variable as abnormal is interpreted as the violation of the influences implied in the estimation of the variable, i.e., the ascending influences in the causal graph. Each influence being associated with a component, this allows one to characterize a set of components constituting a conflict set. The influences of CaEn2 have a "delay" attribute corresponding to a pure delay in the input-output function associated with the influence. This information is used to generate conflict sets whose components are labeled by a time label indicating the date at the latest at which the fault occurred on the component. Diagnoses are obtained from conflict sets by an incremental algorithm that generates hitting sets while managing time labels (Travé-Massuyès and Calderon-Espinoza 2007).

Diagnosis of Hybrid Systems

The works on hybrid systems have been steadily increasing since the pioneering works in the early 2000s (McIlraith et al. 2000). Hybrid systems make it possible to represent double continuous and discrete dynamics that cohabit in many modern systems. Most systems are indeed made up of a set of heterogeneous interconnected

420043_1_En_21_Chapter ✓ TYPESET ☐ DISK ☐ LE ✓ CP Disp.:15/12/2018 Pages: 35 Layout: T1-Standard

components, orchestrated by a supervisor whose commands, of discrete nature, induce different operation modes. Hybrid system modeling as well as associated diagnosis algorithms use continuous and discrete mathematics, so that hybrid systems open a predilection area for integrating methods from the two FDI and DX communities.

The NASA *Livingstone* diagnosis engine (Williams and Nayak 1996), which flew onboard the DS-1 probe, was one of the first to qualify as hybrid. This engine was rooted in the AI model-based diagnosis framework, relying on a model written in propositional logic, and behavioral equations accounting for continuous aspects abstracted in the form of logical relations (qualitative constraints). However, qualitative abstraction imposed *monitors* between the sensors and the model to interpret the actual continuous signals in terms of discrete modalities. The difficulty in deciding proper thresholds and the poor sensitivity of the fault detection procedure led subsequent works to consider true hybrid models, associating differential equation and discrete event models. As proposed in Bayoudh et al. (2008a), Bayoudh and Travé-Massuyès (2014), a hybrid model can be represented in the form of a 6-tuple:

$$S = (\zeta, Q, E, T, K, (q_0, \zeta_0))$$

where:

- ζ is the vector of continuous variables;
- Q is the set of discrete system states, each representing an operating mode of the system;
- E is the set of events corresponding to discrete commands, autonomous mode transitions, or occurrence of faults; events corresponding to autonomous mode transitions are subject to guards that depend on continuous variables;
- $T \subseteq Q \times E \to Q$ is the transition function; it is possible to attach probabilities to the transitions;
- $K = \cup K_i$ is the set of constraints linking the continuous variables, taking the form of differential and possibly algebraic equations modeling the continuous behavior of the system in the different modes $q_i \in Q$;
- $(\zeta_0, q_0) \in \zeta \times Q$ is the initial condition of the hybrid system.

In the hybrid state (ζ, Q), only the discrete state $q_i \in Q$ is representative of the operating mode of the system and provides the diagnosis. However, the evolution of the discrete state is interlinked to the evolution of the continuous state, which is why the problem of diagnosis is often brought back to the problem of estimating the complete hybrid state.

In theory, hybrid estimation presupposes to consider all the sequences of possible modes with the continuous evolution associated with them, which results in exponential complexity. Consequently, many suboptimal methods have been proposed for which we can distinguish the three following families of methods:

- Methods based on *multimode filtering*, rather anchored in the Automatic Control field (Blom and Bar-Shalom 1988; Hofbaur and Williams 2004; Benazera and

Travé-Massuyès 2009), are formulated in a probabilistic framework. They track the different "hypotheses", that is to say the sequences of modes and their associated continuous evolution, over a limited time window and merge the continuous estimates according to a likelihood measure resulting in a *belief state* in the form of a probability distribution over the states at the current time.

- Methods based on *particle filtering* (Arulampalam et al. 2002) are based on sampling and rely on a Bayesian update of the belief state. With enough samples, they approximate the optimal Bayesian estimate but are not well adapted to the problem of the diagnosis because the probabilities of faults are generally very low in comparison with the probabilities of the nominal states of the system.
- Methods that address hybrid aspects in a *dedicated manner* adopt strategies to retrieve the trajectory of the system when it has been discarded due to the approximation of the estimation method (Nayak and Kurien 2000; Benazera and Travé-Massuyès 2003).

Let us note that (Bayoudh et al. 2008b; Vento et al. 2015; Sarrate et al. 2018) propose an alternative approach to complete hybrid diagnosis that only estimates the discrete state, i.e. the operating mode. It combines the parity space approach based on ARRs as defined in Sect. 4.2 for processing the information provided by continuous dynamics with the DES diagnoser method as presented in Sect. 3.4 (Sampath et al. 1995).

Recent works address hybrid system diagnosability integrating a *twin plant* approach as presented in Sect. 3.6 for DES with mode distinguishability methods coming from the FDI community (Grastien et al. 2017). This work is based on abstracting the hybrid automaton model. The continuous dynamics are abstracted remembering only two pieces of information: discernability between modes (when they are guaranteed to generate different observations) and ephemerality (when the system cannot stay forever in a given set of modes). Iterative abstractions can be checked for diagnosability with the standard DES twin plant method that provides a counterexample in case of non-diagnosability. The absence of such a counterexample proves the diagnosability of the original hybrid system. In the opposite case, the counterexample is analyzed to refine the DES. This procedure is referred as a counterexample guided abstraction refinement (CEGAR) scheme. It supports the proposals of Zaatiti et al. (2017, 2018) in which Qualitative Reasoning (see chapter "Qualitative Reasoning" of this volume) is used to compute discrete abstractions. Abstractions as timed automata allow one to handle time constraints that can be captured at a qualitative level.

5 Conclusion

Model-based diagnosis found its formal bases in the 1980s for static systems and in the 1990s with regard to dynamic systems. Since then, developments have been constant and promising, and have in fact become industrialized in several industrial domains such as automotive, aeronautics, space. An important point for the French diagnosis community is the real collaboration of the Automatic Control and AI communities, which brought their respective approaches close together by showing their proximity and their specificities. This has been quite productive on both sides. In the domain of dynamic systems, interest has developed over the last few years on hybrid systems, making it possible to deal with double dynamics, discrete and continuous, and to account for the heterogeneity of current systems. It is a privileged area for the collaborations between the two communities.

One of the current topics is the improvement of the efficiency of existing algorithms to scale up and approach large systems such as those proposed by the DX competition, for instance electronic circuits comprising several thousand components. This involves the use of data structures like BDDs or or very efficient algorithms like SAT, taking into account the structure of the systems. This also involves distributed approaches that divide the problem in a set of problems that are as independent as possible.

Another major pathway concerns the properties of the systems from the diagnosis point of view, namely the in depth study of diagnosability, observability, and repairability for enabling the design of systems which can be monitored, diagnosed and repaired optimally. A last line of work concerns the monitoring of distributed systems for which detection, diagnosis, and return to nominal operating conditions requires good collaboration between methods and tools proposed by the FDI and AI communities. This is also true for planning and decision making.

Finally, as in any situation where model and real world coexist, attention must be paid to the problems linked to the quality and precision of the model, compared to the quality of the information (accuracy, precision, etc.) gathered on the real system, through sensors that can be imperfect and subject to faults. For all these developments, it can be noted that this involves dialogue and co-operation with researchers from many fields, in particular those from the AI community. This is obviously a challenge but also an opportunity for reciprocal fertilization.

For detailed references on the topic of diagnosis, it is best to consult the proceedings of the international conference DX (Principles of diagnosis), which brings together every year researchers in the field (DX 2018).

References

Aghasaryan A, Fabre E, Benveniste A, Boubour R, Jard C (1997) A Petri net approach to fault detection and diagnosis in distributed systems. II. Extending Viterbi algorithm and HMM techniques to Petri nets. In: 36th IEEE conference on decision and control, San Diego (CA), États-Unis, pp 726–731

Armengol J, Bregon A, Escobet T, Gelso E, Krysander M, Nyberg M, Olive X, Pulido B, Travé-Massuyès L (2009) Minimal structurally overdetermined sets for residual generation: a comparison of alternative approaches. In: 7th IFAC symposium on fault detection, supervision and safety of technical processes, Barcelone, Espagne, pp 1480–1485

Arulampalam MS, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188

Bayoudh M, Travé-Massuyès L (2014) Diagnosability analysis of hybrid systems cast in a discrete-event framework. Discret Event Dyn Syst 24(3):309–338

Bayoudh M, Travé-Massuyès L, Olive X (2008a) Coupling continuous and discrete event system techniques for hybrid system diagnosability analysis. In 18th European conference on artificial intelligence including prestigious applications of intelligent, Patras, Grèce. IOS Press, pp 219–223

Bayoudh M, Travé-Massuyès L, Olive X (2008b) Hybrid systems diagnosis by coupling continuous and discrete event techniques. In: Proceedings of the IFAC world congress, Seoul, Korea, pp 7265–7270

Benazera E, Travé-Massuyès L (2003) The consistency approach to the on-line prediction of hybrid system configurations. In: Analysis and design of hybrid systems 2003 (ADHS 03): a proceedings volume from the IFAC Conference, St. Malo, Brittany, France, 16–18 June 2003. Elsevier Science, pp 241–246

Benazera E, Travé-Massuyès L (2009) Set-theoretic estimation of hybrid system configurations. IEEE Trans Syst Man Cybern. Part B Cybern: Publ IEEE Syst Man Cybern Soc 39(6):1277–1291

Besnard P, Cordier M-O (1994) Explanatory diagnoses and their characterization by circumscription. Ann Math Artif Intell 11:75–96

Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2015) Diagnosis and fault-tolerant control, 3rd edn. Springer, Berlin

Blanke M, Kinnaert M, Schröder J, Lunze J, Staroswiecki M (2003) Diagnosis and fault-tolerant control. Springer, Berlin

Blom H, Bar-Shalom Y (1988) The interacting multiple model algorithm for systems with Markovian switching coefficients. IEEE Trans Autom Control 33:780–783

Brusoni V, Console L, Terenziani P, Dupré DT (1998) A spectrum of definitions for temporal model-based diagnosis. Artif Intell 102:39–79

Carle P, Choppy C, Kervarc R (2011) Behaviour recognition using chronicles. In: 2011 fifth international conference on theoretical aspects of software engineering, pp 100–107

Chatti N, Ould-Bouamama B, Gehin A-L, Merzouki R (2014) Signed bond graph for multiple faults diagnosis. Eng Appl Artif Intell 36:134–147

Chittaro L, Ranon R (2004) Hierarchical model-based diagnosis based on structural abstraction. Artif Intell 1–2:147–182

Chow E, Willsky A (1984) Analytical redundancy and the design of robust failure detection systems. IEEE Trans Autom Control 29(7):603–614

Cimatti A, Pecheur C, Cavada R (2003) Formal verification of diagnosability via symbolic model checking. In: Proceedings of the 18th international joint conference on artificial intelligence IJCAI'03, Acapulco, Mexique, pp 363–369

Console L, Picardi C, Ribaudo M (2002) Process algebra for systems diagnosis. Artif Intell 142:19–51

Console L, Torasso P (1990) Hypothetical reasoning in causal models. Int J Intell Syst 5(1):83–124

Console L, Torasso P (1991) A spectrum of logical definitions of model-based diagnosis. Comput Intell 7:133–141

Contant O, Lafortune S, Teneketzis D (2004) Diagnosis of intermittent faults. Discret Event Dyn Syst: Theory Appl 14(2):171–202

Cordier M, Dague P, Lévy F, Montmain J, Staroswiecki M, Travé-Massuyès L (2004) Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. IEEE Trans Syst Man Cybern Part B 34(5):2163–2177

Cordier M-O (1998) When abductive diagnosis fails to explain too precise observations: an extended spectrum of model-based diagnosis definitions based on abstracting observations. In: Proceedings of DX'98, Cape Cod (MA), États-Unis, pp 24–31

Cordier M-O, Pencolé Y, Travé-Massuyès L, Vidal T (2007) Self-healability = diagnosability + repairability. In: 18th international workshop on principles of diagnosis, Nashville, Tennessee, United States, pp 251–258
Cordier M-O, Thiébaux S (1994) Event-based diagnosis for evolutive systems. In: 5th international workshop on principles of diagnosis (DX-94), New Palz (NY), États-Unis, pp 64–69
Cordier M-O, Travé-Massuyès L, Pucel X (2006) Comparing diagnosability in continuous and discrete-event systems. In: 17th international workshop on principles of diagnosis (DX06), Burgos, Espagne, pp 55–60
Dague P, Jehl O, Taillibert P (1990) An interval propagation and conflict recognition engine for diagnosing continuous dynamic systems. In: Expert systems in engineering, pp 16–31
Dague P, Travé-Massuyès L (2004) Raisonnement causal en physique qualitative. Intellectica 38:247–290
De Kleer J (1992) Focusing on probable diagnosis. Readings in model-based diagnosis. Morgan Kaufmann, San Mateo
De Kleer J (2006) Improving probability estimates to lower diagnostic costs. In: 17th international workshop on principles of diagnosis (DX06), Burgos, Espagne, pp 55–60
De Kleer J, Williams B (1987) Diagnosing multiple faults. Artif Intell 32(1):97–130
Debouk R, Lafortune S, Teneketzis D (2002) Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discret Event Dyn Syst: Theory Appl 10(1–2):33–86
Dousson C (1996) Alarm driven supervision for telecommunication networks: II -On line chronicle recognition. Annales des Télécommunications 51(9–10):501–508
Dubuisson B (2001) Automatique et statistiques pour le diagnostic. Hermes Science Europe Ltd
DX (2018) Proceedings of the 0th to 29th international workshop on principles of diagnosis, 1989–2018
Fabre E, Benveniste A, Haar S, Jard C (2005) Distributed monitoring of concurrent and asynchronous systems. Discret-Event Dyn Syst: Theory Appl 15(1):33–84
Feldman A, Provan G, Van Gemund A (2009) FRACTAL: efficient fault isolation using active testing. In: Proceedings of the international joint conference on artificial intelligence (IJCAI'09), Pasadena (CA), États-Unis, pp 778–784
Friedrich G, Gottlob G, Nejdl W (1994) Formalizing the repair process - extended report. Ann Math Artif Intell 11(1–4):187–201
Gao Z, Cecati C, Ding SX (2015) A survey of fault diagnosis and fault-tolerant techniques part i: fault diagnosis with model-based and signal-based approaches. IEEE Trans Ind Electron 62(6):3757–3767
Gentil S, Montmain J, Combastel C (2004) Combining FDI and AI approaches within causal-model-based diagnosis. IEEE Trans Syst Man Cybern Part B 34(5):2207–2221
Gertler J (1998) Fault detection and diagnosis in engineering systems. Marcel Deker, New York
Gougam H-E, Pencolé Y, Subias A (2017) Diagnosability analysis of patterns on bounded labeled prioritized Petri nets. J Discret Event Dyn Syst: Theory Appl 27(1):143–180
Grastien A, Cordier M-O, Largouët C (2005) Automata slicing for diagnosing discrete-event systems with partially ordered observations. In: 9th congress of the Italian association for artificial intelligence, Milan, Italie, pp 270–281
Grastien A, Travé-Massuyès L, Puig V (2017) Solving diagnosability of hybrid systems via abstraction and discrete event techniques. IFAC-PapersOnLine 50(1):5023–5028
Grastien Al, Anbulagan An (2013) Diagnosis of discrete event systems using satisfiability algorithms: a theoretical and empirical study. IEEE Trans Autom Control (TAC) 58(12):3070–3083
Greiner R, Smith B, Wilkerson R (1989) A correction to the algorithm in Reiter's theory of diagnosis. Artif Intell 41:79–88
Hofbaur MW, Williams BC (2004) Hybrid estimation of complex systems. IEEE Trans Syst, Man, Cybern-Part B: Cybern 34(5):2178–2191
Jéron T, Marchand H, Pinchinat S, Cordier M-O (2006) Supervision patterns in discrete event systems diagnosis. In: Workshop on discrete event systems, WODES'06, Ann-Arbor (MI), États-Unis, pp 262–268

Jiang S, Huang Z, Chandra V, Kumar R (2001) A polynomial time algorithm for diagnosability of discrete event systems. IEEE Trans Autom Control 46(8):1318–1321

Jiroveanu G, Boel R (2006) A distributed approach for fault detection and diagnosis based on time Petri nets. Math Comput Simul 70(5–6):287–313

KanJohn P, Grastien A (2008) Local consistency and junction tree for diagnosis of discrete-event systems. In: European conference on artificial intelligence (ECAI-08). Patras, Grèce, pp 209–213

Kleer J, Mackworth A, Reiter R (1992) Characterizing diagnoses and systems. Artif Intell 56(2–3):197–222

Kramer MA, Palowitch BL (1987) A rule-based approach to fault diagnosis using the signed directed graph. AIChE J 33(7):1067–1078

Krysander M, Åslund J, Nyberg M (2008) An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. IEEE Trans Syst, Man, Cybern-Part A: Syst HumS 38(1):197–206

Lamperti G, Zanella M (2003) Diagnosis of active systems. Kluwer Academic Publishers, Dordrecht

Loiez E, Taillibert P (1997) Polynomial temporal band sequences for analog diagnosis. In: IJCAI-97: proceedings of the fifteenth international joint conference on artificial intelligence, Nagoya, Japon, pp 474–479

Lunze J (1994) Qualitative modelling of linear dynamical systems with quantized state measurements. Automatica 30(3):417–431

Marchand H, Rozé L (2002) Diagnostic de pannes sur des systèmes à événements discrets : une approche à base de modèles symboliques. In: 13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle. Angers, France, pp 191–200

McCarthy J (1986) Applications of circumscription to formalizing common-sense knowledge. Artif Intell 28:89–116

McIlraith S, Biswas G, Clancy D, Gupta V (2000) Hybrid systems diagnosis. Lecture notes in computer science, pp 282–295

Nayak P, Kurien J (2000) Back to the future for consistency-based trajectory tracking. In: Proceedings of AAAI-2000, Austin (TX), États-Unis, pp 370–377

Nejdl W, Bachmayer J (1993) Diagnosis and repair iteration planning versus n-step look ahead planning. In: 4th international workshop on principles of diagnosis, Aberystwyth, Royaume Uni

Patton R, Chen J (1991) A re-examination of the relationship between parity space and observer-based approaches in fault diagnosis. Eur J Diagn Saf Autom 1(2):183–200

Pencolé Y (2004) Diagnosability analysis of distributed discrete event systems. In: European conference on artificial intelligence (ECAI'04). Valence, Espagne, pp 43–47

Pencolé Y, Cordier M-O (2005) A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. Artif Intell 164:121–170

Pencolé Y, Schumann A, Kamenetsky D (2006) Towards low-cost fault diagnosis in large component-based systems. In: 6th IFAC symposium on fault detection, supervision and safety of technical processes, Pékin, Chine, pp 1473–1478

Pencolé Y, Steinbauer G, Mühlbacher C, Travé-Massuyès L (2018) Diagnosing discrete event systems using nominal models only. In: 28th international workshop on principles of diagnosis, Brescia, Italy, pp 169–183

Pencolé Y, Subias A (2018) Diagnosis of supervision patterns on bounded labeled petri nets by model checking. In: 28th international workshop on principles of diagnosis, Brescia, Italy, pp 184–199

Peng Y, Reggia JA (1990) Abductive inference models for diagnsotic problem-solving. Springer, Berlin

Pons R, Subias A, Travé-Massuyès L (2015) Iterative hybrid causal model based diagnosis: application to automotive embedded functions. Eng Appl Artif Intell 37:319–335

Poole D (1989) Normality and faults in logic-based diagnosis. In: IJCAI, pp 1304–1310

Provan G (2002) On the diagnosability of decentralized, timed discrete event systems. In: 41st IEEE conference on decision and control, Las Vegas (NV), États-Unis, pp 405–410

Pulido B, Gonzalez C (2004) Possible conflicts: a compilation technique for consistency-based diagnosis. IEEE Trans Syst, Man, Cybern, Part B 34(5):2192–2206

Reggia JA, Nau D, Wang Y (1983) Diagnostic expert systems based on a set covering model. Int J Man-Mach Stud 19:437–460

Reiter R (1987) A theory of diagnosis from first principles. Artif Intell 32(1):57–95

Ribot P, Pencolé Y, Combacau M (2008) Design requirements for the diagnosability of distributed discrete event systems. In: 19th international workshop on principles of diagnosis. Blue Mountains, Nouvelle-Galles du Sud, Australie, pp 347–354

Rozé L, Cordier M-O (2002) Diagnosing discrete-event systems: extending the “diagnoser approach” to deal with telecommunication networks. Discret-Event Dyn Syst: Theory Appl 12(1):43–81

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1995) Diagnosability of discrete event system. IEEE Trans Autom Control 40(9):1555–1575

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1996) Failure diagnosis using discrete-event models. IEEE Trans Control Syst Technol 4(2):105–124

Sarrate R, Puig V, Travé-Massuyès L (2018) Diagnosis of hybrid dynamic systems based on the behavior automaton abstraction. In: Fault diagnosis of hybrid dynamic and complex systems. Springer, Berlin, pp 243–278

Schumann A, Pencolé Y (2007) Scalable diagnosability checking of event-driven system. In: Proceedings of the twentieth international joint conference on artificial intelligence (IJCAI07), Hyderabad, Inde, pp 575–580

Schumann A, Pencolé Y, Thiébaux S (2004) Diagnosis of discrete-event systems using binary decision diagrams. In: Proceedings of the internationalworkshop on principles of diagnosis (DX’04), Carcassonne, France, pp 197–202

Schumann A, Pencolé Y, Thiébaux S (2010) A decentralised symbolic diagnosis approach. In: 19th European conference on artificial intelligence (ECAI-10). IOS Press, Lisbonne, Portugal, pp 99–104

Siddiqi S, Huang J (2010) New advances in sequential diagnosis. In: Proceedings of the twelfth international conference on the principles of knowledge representation (KR’10), Toronto, Canada, pp 17–25

Staroswiecki M, Comtet-Varga G (2001) Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. Automatica 37(5):687–699

Su X, Grastien Al (2013) Diagnosis of discrete event systems by independent windows. In: 24th international workshop on principles of diagnosis (DX-13), Jerusalem, Israel, pp 148–153

Su X, Grastien Al, Pencolé Ya (2014) Window-based diagnostic algorithms for discrete event systems: what information to remember. In: 25th international workshop on principles of diagnosis (DX-14)

Su X, Zanella M, Grastien A (2016) Diagnosability of discrete-event systems with uncertain observations. In: 25th international joint conference on artificial intelligence (IJCAI-16), pp 1265–1271

Sun Y, Weld DS (1993) A framework for model-based repair. In: 11th national conference on artificial intelligence, Washington, D.C., États-Unis, pp 182–187

Ten Teije A, Van Harmelen F (1994) An extended spectrum of logical definitions for diagnostic systems. In: Proceedings of DX-94 Fifth International Workshop on Principles of Diagnosis, New Paltz (NY), États-Unis, pp 334–342

Torta G, Torasso P (2003) Automatic abstraction in component-based diagnosis driven by system observability. In: Proceedings of the 18th international joint conference on artificial intelligence - IJCAI03, Mexique, Acapulco, pp 394–400

Travé-Massuyès L (2014) Bridging control and artificial intelligence theories for diagnosis: a survey. Eng Appl Artif Intell 27:1–16

Travé-Massuyès L, Calderon-Espinoza G (2007) Timed fault diagnosis. In: Proceedings of the IEEE European control conference (ECC-07), Kos, Grèce, pp 2272–2279

Travé-Massuyès L, Dague P (2003) Modèles et raisonnements qualitatifs. Hermes sciences

Travé-Massuyès L, Escobet T, Milne R (2001) Model-based diagnosability and sensor placement application to a frame 6 gas turbine subsystem. In: Proceedings of the seventeenth international joint conference on artificial intelligence, IJCAI'01, vol 1, pp 551–556

Travé-Massuyès L, Pons R, Tornil S, Escobet T (2001) The CA-En diagnosis system and its automatic modelling method. Computación y Sistemas 5(2):128–143

Vento J, Travé-Massuyès L, Puig V, Sarrate R (2015) An incremental hybrid system diagnoser automaton enhanced by discernibility properties. IEEE Trans Syst, Man, Cybern: Syst 45(5):788–804

Washio T, Motoda H, Niwa Y (1999) Discovering admissible model equations from observed data. In Proceeding of IJCAI99: sixteenth international joint conferenceon artificial intelligence, vol 2, Stockholm, Suède, pp 772–779

Weld D, De Kleer J (1989) Readings in qualitative reasoning about physical systems. Morgan Kaufmann Publishers Inc

Williams BC, Nayak P (1996) A model-based approach to reactive self-configuring systems. In: Proceedings of the 13th national conference on artificialintelligence (AAAI-96), Portland (OR), États-Unis, pp 971–978

Ye L, Dague P (2012) A general algorithm for pattern diagnosability of distributed discrete event systems. In: ICTAI - 24th international conference ontools with artificial intelligence, Athènes, Greece

Ye L, Dague P (2017) An optimized algorithm of general distributed diagnosability analysis for modular structures. IEEE Trans Autom Control 62(4):1768–1780

Yoo T, Lafortune S (2002) Polynomial-time verification of diagnosability of partially-observed discrete-event systems. IEEE Trans Autom Control 47(9):1491–1495

Zaatiti H, Ye L, Dague P, Gallois J-P (2017) Counter example guided abstraction refinement for hybrid systems diagnosability analysis. In: 28th internationalworkshop on principles of diagnosis (DX-17)

Zaatiti H, Ye L, Dague P, Gallois J-P, Travé-Massuyès L (2018) Abstractions refinement for hybrid systems diagnosability analysis. In: Diagnosability, security and safety of hybrid dynamic and cyber-physical systems. Springer, Berlin, pp 279–318