HAL
open science

Bridging Control and Artificial Intelligence Theories for Diagnosis: A survey

Louise Travé-Massuyès

To cite this version:

Louise Travé-Massuyès. Bridging Control and Artificial Intelligence Theories for Diagnosis: A survey. Engineering Applications of Artificial Intelligence, 2013, 27 (27), p.1-16. 10.1016/j.engappai.2013.09.018 . hal-01131188

HAL Id: hal-01131188

https://hal.science/hal-01131188v1

Submitted on 21 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bridging Control and Artificial Intelligence Theories for Diagnosis: A survey

Louise Travé-Massuyès*

Abstract

Diagnosis is the process of identifying or determining the nature and root cause of a failure, problem, or disease from the symptoms resulting from selected measurements, checks or tests. The different facets of this problem and the wide spectrum of classes of systems make it interesting to several communities and require bridging several theories. Diagnosis is actually a functional fragment in fault management architectures and it must smoothly interact with other functions. This paper presents diagnosis as it is understood in the Control and Artificial Intelligence fields, and exemplifies how different theories of these fields can be synergistically integrated to provide better diagnostic solutions and to achieve improved fault management in different environments[1].

Keywords— model based diagnosis, data based diagnosis, abstractions, fault management architectures.

1 Introduction

The goal of diagnosis is to identify the possible causes explaining a set of observed symptoms. A set of concomitant tasks contribute to this goal and the following three tasks are commonly identified:

*CNRS, LAAS, Univ de Toulouse, 7 avenue du colonel Roche, F-31400 Toulouse, France louise@laas.fr

[1]The author acknowledges the *8th IFAC Symposium on Fault Detection Supervision and Safety of Technical Process SAFEPROCESS'12*, August 29-31 2012, Mexico City, Mexico, for recommending the symposium version of this paper for publication in the IFAC Journal on Engineering Applications of Artificial Intelligence.

- *fault detection*, which aims at discriminating normal system states from abnormal ones, i.e. states which result from the presence of a fault,
- *fault isolation*, also called *fault localization*, whose goal is to point at the faulty components of the system,
- *fault identification*, whose output is the type of fault and possibly the model of the system impacted by this fault.

Faced with the diversity of systems and different views of the above problems, several scientific communities have addressed these tasks and contributed with a large spectrum of methods. The Signal Processing, Control and Artificial Intelligence (AI) communities are leading actors in this field.

Diagnosis is carried out from the signals that permit efficient fault detection towards the upper levels of supervision that call for qualitative interpretations. Proposing relevant abstractions to interpret the available signals is hence a key issue.

Signal processing provides specific contributions in the form of statistic algorithms for detecting changes in signals, hence detecting faults. This remains out of the scope of this paper and has been surveyed in several reference books and papers (Basseville, 1988; Basseville and Nikiforov, 1993; Basseville et al., 2004; Fillatre and Nikiforov, 2007; Fouladirad et al., 2008).

Interfaces between continuous signals and their abstract interpretations, in symbolic or event-based form, implement qualitative interpretations of the signals that are required for supervision. To do that, discrete formalisms borrowed from AI find a natural link with continuous models from the Control community. These two communities have their own model-based diagnosis track :

- the FDI (Fault Detection and Isolation) track, whose foundations are based on engineering disciplines, such as control theory and statistical decision making,
- the DX (Diagnosis) track, whose foundations are derived from the fields of logic, combinatorial optimization, search and complexity analysis.

In the last decade, there has been a growing number of researchers in both communities who have tried to understand and bridge FDI and DX approaches to build better, more robust and effective diagnostic systems.

Data-based diagnosis approaches based on machine learning techniques, such as pattern recognition (Fukunaga, 1990; Denoeux et al., 1997), are present in both the Control and AI communities and complement model-based approaches to provide solutions to a variety of diagnostic problems

where the difficulty arises from the scarce nature of the instrumentation or, conversely, from the massive amounts of data to be interpreted to extract hidden knowledge. Interesting bridges also arise when we consider data-based and model-based approaches.

Other bridges can be found when considering that diagnosis is not a goal per se but a component in fault management architectures. It takes part in the solutions produced for tasks such as failure-mode-and-effects analysis, sensor placement, on-board recovery, condition monitoring, maintenance, repair and therapy planning, and prognosis. The contribution of diagnosis in such architectures requires close links with decision tasks such as control and planning and calls for innovative integrations.

In this paper, different facets of diagnosis investigated in the Control or the AI fields are discussed. While (Venkatasubramanian et al., 2003a,b,c) provide three interesting surveys of the different approaches that exist in these fields, this paper aims at reporting the works that integrate approaches of both sides, hence creating "bridges". In particular, the concepts and results of the FDI and DX tracks are put in correspondence and the lessons learned from this comparative analysis are pointed out. Causal model-based diagnosis is presented as a typical example of integration of FDI and DX theories, in which fault detection is implemented along an FDI approach and fault isolation calls for the logical DX framework. Hybrid model-based diagnosis is then used to illustrate several interesting bridges, in particular, how FDI estimation schemes can be combined with search algorithms rooted in AI to achieve hybrid state tracking efficiently. Combining FDI estimation filters with the logical DX theory is also illustrated. Finally, it is shown that the hybrid model-based diagnosis problem can also find a solution by combining the FDI approach with the so-called diagnoser approach of the discrete event systems (DES) field. Subsequently, learning the models that support diagnosis reasoning is shown to be a rich field for bridging theories. It has been intensively investigated for continuous model identification for which regression analysis is essential. Learning discrete event models calls for other bridges, which are illustrated with chronicle learning. Finally, diagnosis is discussed in relation with theories that participate to provide global solutions to fault management problems. On one hand, autononous architectures exemplify the integration of diagnosis, control and planning. On the other hand, it is shown how diagnosis can enhance prognosis in condition maintenance architectures.

The paper is organized as follows. After the introduction section, section 2 first presents a brief overview of the approaches proposed by the FDI and DX model-based diagnosis communities. Although quite commonplace, this overview is necessary because it provides the basic concepts and principles

that form the foundations of any diagnosis method. It is followed by the comparison of the concepts and techniques used by these communities and the lessons learned from this comparative analysis. Section 3 is concerned with the trends that integrate and take advantage of techniques from both sides, in particular causal model based diagnosis in subsection 3.1 and diagnosis of hybrid systems in subsection 3.2. Section 4 then raises the problem of obtaining the models supporting diagnosis reasoning and discusses bridges that can contribute to learning them in an automated manner. Section 5 widens the scope of diagnosis and is concerned with diagnosis as a component of fault management architectures, discussing the several links with control and planning. Finally, section 6 concludes the paper.

2 DX and FDI model-based diagnosis bridge

The FDI and DX streams both consider the diagnosis problem from a *system* point of view, which results in significant overlaps. Even the name of the two tracks are the same : *Model-Based Diagnosis* (MBD).

The diagnosis principles are the same, although each community has developed its own concepts and methods, guided by different modeling paradigms and solvers. FDI relies on analytical models, linear algebra, and non-linear system theory whereas DX takes its foundations in logic. In the 2000s, catalyzed by the *"Bridging AI and Control Engineering model-based diagnosis approaches"* group, known as the BRIDGE group (BRIDGE), within the European Network of Excellence MONET II (MONET), and its French counterpart, the *"Intégration de Méthodes Alliant Automatique et IA"* group, known as the IMALAIA group, supported by GDR MACS (GDR MACS), GDR I3 (GDR-I3), as well as AFIA (AFIA), there were more and more researchers who tried to understand and synergistically integrate methods from the two tracks to propose more efficient diagnostic solutions. This collaboration launched several events :

- a BRIDGE Workshop in 2001 in the framework of DX'01, the *12th International workshop on Principles of Diagnosis*, Sansicario, Via Lattea, Italy, 5-9 Mars 2001 (DX'01, 2001).

- the co-location, in Washington DC (USA), of the two main events of the FDI and the DX communities, namely the *IFAC International Symposium on Fault Detection, Supervision and Safety for Technical Processes* SAFEPROCESS'03 and the *International Workshop on Principles of Diagnosis* DX'03, including a BRIDGE Workshop in the form of a join day.

These events were followed by the publication of a special issue of the IEEE SMC Transactions, Part B, on the topic *"Diagnosis of Complex Systems: Bridging the methodologies of the FDI and DX Communities"* in 2004 (Biswas et al., 2004). The *BRIDGE track* was launched and paved its way until today. Other events followed like the two invited sessions *"AI methods for Model-based Diagnosis"* and *"Bridge between Control Theory and AI methods for Model-based Diagnosis"*, recently organized in the framework of the *7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes* SAFAPROCESS'09, Barcelona, Spain, 30 July-3 August 2009.

The next subsections first summarize the foundations of the FDI and DX approaches, then proceed with a comparative analysis that allows us to draw some practical assessments in the form of lessons learned. The lessons summarize the respective strengths and weaknesses of the two approaches and provide the guidelines that drive the proposals combining the two approaches.

2.1 Brief overview of FDI approaches

The detection and diagnosis methods of the FDI community rely on behavioral models that establish the constraints between system inputs $u \in U$ and outputs $y \in Y$, gathered in the set of measurable variables Z, and the system internal states defining the set of unknown variables X. The variables $z \in Z$ and $x \in X$ are functions of time. The typical model may be formulated in the temporal domain, then known as a *state-space model* :

$$\begin{array}{ll} BM: & dx/dt = f(x(t), u(t), \theta) \\ OM: & y(t) = g(x(t), u(t), \theta). \end{array} \tag{1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, $u(t) \in \mathbb{R}^{n_u}$ is the input vector and $y(t) \in \mathbb{R}^{n_p}$ is the output vector. $\theta \in \mathbb{R}^{n_\theta}$ is a constant parameter vector. The components of f and g are real functions over $\mathbb{R}$. BM is the behavioral model and OM is the observation model. The whole system model is noted $SM(z, x)$, like in (Krysander et al., 2008), and assumed noise-free. The equations of $SM(z, x)$ may be associated to components but this information is not represented explicitly. The models can also be formulated in the frequency domain, for instance in the form of transfer functions in the linear case.

Models are used in three families of methods:

- the methods based on *parameter estimation* that focus on the value of parameters as representing physical features of the system,

- the methods based on *state estimation*, like observers and filters, that rely on the estimation of unknown variables,
- the methods based on the *parity space* that rely on the elimination of unknown variables.

The books (Gertler, 1998), (Blanke et al., 2003), (Dubuisson, 2001), (Patton et al., 1989) provide excellent surveys, which cite the original papers that the reader is encouraged to consult. The paper (Venkatasubramanian et al., 2003c) also provides a quite comprehensive survey. The equivalence between observers, parity space and paramater estimation has been proved in the linear case (Patton and Chen, 1991).

The concept central to FDI methods is the concept of *residual* and one of the main problems is to *generate residuals*. Let's consider the model $SM(z, x)$ given by (1), then $SM(z, x)$ is said to be consistent with an observed trajectory z, or simply *consistent with measurements* z, if there exists a trajectory of x such that the equations of $SM(z, x)$ are satisfied.

Definition 1 (Residual generator for $SM(z, x)$) *A system that takes as input a sub-set 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 system model is consistent with measurements, the residuals tend to zero as t tends to infinity, otherwise some residuals may be different from zero. In practice, the residuals never happen to be exactly *zero* because the noise-free assumption adopted for (1) is never met. Evaluating the residuals and assigning them a Boolean value – 0 or non 0 – requires statistical tests that account for the statistical characteristics of noise (Basseville and Nikiforov, 1993; Dubuisson, 2001). The residuals are often optimized to be robust to disturbancies (Qiu and Gertler, 1993) and to take into account uncertainties (Adrot et al., 1999).

The methods based on parameter estimation are used for linear as well as non linear systems (Pouliezos et al., 1985). Fault detection is achieved by comparing the estimated parameter values to the nominal values. With these methods, fault detection, isolation, and identification are achieved at once, provided that model parameters can be put in correspondence with physical parameters. The results rely on the identifiability of the system as explained in (Grewal and Glover, 1976) and (Jauberthie et al., 2011) in the set-membership framework.

The methods based on state estimation take the form of *observers* or optimized *filters*, such that the Kalman filter, and provide an estimation of

the state of the system. Specific filter architectures are often designed to structure the residuals, i.e. to sensitize different residuals for different faults, and isolate the faults (Frank, 1994). Numerous diagnosis solutions rely on state estimation, particularly for hybrid systems. These works are surveyed in section 3.2. 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 the parity space (Chow and Willsky, 1984) generate residuals from relations that are inferred from the system model. These relations, called *Analytical Redundancy Relations* (ARR), are determined off-line.

Definition 2 (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.*

ARRs are constraints that only involve measured input and output variables and their derivatives. For linear systems, ARRs are obtained eliminating unknown state variables by linear projection on a particular space, called the *parity pace* (Chow and Willsky, 1984). An extension to non-linear systems is proposed in (Staroswiecki and Comtet-Varga, 2001). On the other hand, the structural approach (Staroswiecki and Declerck, 1989; Armengol et al., 2009) is an interesting approach because it allows one to obtain, for linear or non-linear systems, the just overdeterminated sets of equations from which ARRs can be derived.

Every ARR can be put in the form $r(t) = 0$, where $r(t)$ is the *residual*. If the behavior of the system satisfies the model constraints, then the residuals are zero because the ARRs are satisfied. Otherwise, some of them may be different from zero when the corresponding ARRs are violated. Given a set of n residuals, a *theoretical fault signature* $FS_j = [s_{1j}, s_{2j}, \dots, s_{nj}]$ given by the Boolean evaluation of each residual is associated to each fault F_j. The *signature matrix* is then defined as follows.

Definition 3 (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, i.e. $FS = [FS_1, FS_2, \dots, FS_{n_f}]$.*

Diagnosis is achieved by comparing the observed signature, i.e. the Boolean residual values obtained from the actual measurements, and the theoretical signatures of the n_f faults.

2.2 Brief overview of the DX logical diagnosis theory

In the model-based logical diagnosis theory of DX as proposed by (Reiter, 1987; Kleer et al., 1992), the description of the system is driven by components and relies, in its original version, on first order logic. A system is given by a tuple $(SD, COMPS, OBS)$ where:

- SD is the *system description* in the form of a set of first order logic formulas with equality,
- $COMPS$ represents the *set of components* of the system given by a finite set of constants,
- OBS is a set of first order formulas, which represent the *observations*.

SD uses the specific predicate AB, meaning *abnormal*. Applied to a component c of $COMPS$, $\neg AB(c)$ means that c is normal and $AB(c)$ that c is faulty. For instance, the model of a two inputs adder would be given by :

$$\neg AB(x) \wedge ADD(x) \Rightarrow Output(x) := Input1(x) + Input2(x) \qquad (2)$$

Definition 4 (Diagnosis) *A diagnosis for the system* $(SD, COMPS, OBS)$ *is a set* $\Delta \subseteq COMPS$ *such that* $SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPS - \Delta\}$ *is satisfiable.*

The above definition means that the assumption stating that the components of Δ are faulty and all the others are normal is consistent with the observations OBS and the system description SD. A diagnosis hence consists in the assignment of a mode, normal or faulty [2], to each component of the system, which is consistent with the model and the observations.

Definition 5 (Minimal diagnosis) *A* minimal diagnosis *is a diagnosis* Δ *such that* $\forall \Delta' \subset \Delta$, Δ' *is not a diagnosis.*

To obtain the set of diagnoses, it is usual to proceed in two steps, basing the first step on the concept of *conflict* introduced in (Reiter, 1987) and later extended in (Kleer et al., 1992). The original definition, that we call *R-conflict*, i.e. conflict in the sense of Reiter, is the following :

Definition 6 (R-conflict and minimal R-conflict) *An* R-conflict *is a set* $\mathscr{C} \subseteq COMPS$ *such that the assumption that all the components of* $\mathscr{C}$ *are normal is not consistent with* SD *and* OBS. *A* minimal R-conflict *is an R-conflict that does not contain any other conflict.*

[2] This framework has been extended to fault models in (Kleer et al., 1992).

The set of diagnoses can be generated from the set of conflicts. (Reiter, 1987) proved that minimal diagnoses are given by the *hitting sets* [3] of the set of minimal R-conflicts. An algorithm based on the construction of a tree, known as the HS-tree, was originally proposed by (Reiter, 1987).

The parsimony principle indicates that preference should be given to minimal diagnoses. Another reason why minimal diagnoses are important is because in many cases, they characterize the whole set of diagnoses. In other words, all the supersets of minimal diagnoses are diagnoses. (Kleer et al., 1992) provided the conditions for this to be true by extending the definition of an R-conflict to a disjunction of AB-literals, $AB(c)$ or $\neg AB(c)$, containing no complementary pair, entailed by $SD \cup OBS$. Then, a *positive conflict* is a conflict for which all of its literals are positive and one can identify a positive conflict with an R-conflict (Reiter, 1987) as defined above.

Diagnoses are characterized by minimal diagnoses if and only if all minimal conflicts are positive (Kleer et al., 1992). Unfortunately, only sufficient conditions exist on the syntactic form of SD and OBS. One of those is that the clause form of $SD \cup OBS$ only contains positive AB-literals. This is verified, for instance, if all sentences of SD are of the same form as (2), which means that only necessary conditions of correct behavior are expressed.

2.3 Lessons learned from comparing the FDI and DX approaches

This section summarizes the results of (Cordier et al., 2004) obtained from the comparative analysis of the DX approach and the parity space FDI approach presented in the two preceding sections. It presents the concept of *ARR support* that is the articulation of the two approaches and gives rise to the important properties of *ARR–d–completeness* and *ARR–i–completeness*. It then discusses the lessons learned, outlining the different hypotheses adopted by the two approaches and sheding light on their respective strengthes and drawbacks.

Frst, it should be noticed that the modeling paradigm of FDI does not make explicit use of the concept of component. The behavioral model (1) describes the system as a whole. On the contrary, the DX approach models every component independently, and specifies the structure of the system, i.e. how the different components are connected. Another important difference is that the assumption of correct behavior is represented explicitly in SD thanks to the predicate AB. If $\mathcal{F}$ is a formula describing the normal behavior

[3] The hitting sets of a collection of sets are given by the sets that intersect every set of the collection.

of a component, SM only contains $\mathcal{F}$ whereas SD contains the formula $\neg AB(c) \Rightarrow \mathcal{F}$.

The comparison of the two approaches is only possible if the models on both sides represent the same system and the observations/measurements capture the same reality. This is formalized by the *System Representation Equivalence* (SRE) property introduced in (Cordier et al., 2004), which requires that SM is obtained from SD by setting to *false* all the occurrences of the predicate AB. It is also assumed that the same observation language is used, i.e. OBS is a conjunction of equality relations, which assign a value to every measured variable. In addition, the faults refer to the same entities assumed to be components, without loss of generality.

The comparison is based on the concept of signature matrix FS, as defined in definition 3. FS crosses ARRs in rows and faults/components in columns (faults are univocally associated to components). The comparison also relies on the concept of *ARR Support*.

Definition 7 (ARR Support) *Consider ARR_i to be an ARR for $SM(z, x)$, then the* support *of ARR_i, noted supp(ARR_i), is the set of components $\{c_j\}$ (columns of the signature matrix FS) whose corresponding matrix cells FS_{ij} are non zero on the ARR_i line.*

The support of an ARR of the form $r(z, \dot{z}, \ddot{z}, \dots) = 0$ indicates the set of components whose models, or submodels, are involved in the obtention of the relation $r(z, \dot{z}, \ddot{z}, \dots) = 0$. The equations of the model $SM(z, x)$ can indeed be partitioned in component models and every equation of $SM(z, x)$ can be labelled as being part of the model of some component. Let $SM(c)$ denote the subset of equations defining the model of a component $c \in COMPS$ and $SM(C) = \bigcup_{c \in C} SM(c)$ the subset of equations corresponding to $C \subseteq COMPS$.

Let us now introduce two completeness properties, which refer to "detectability" indicated by a "d", and to "isolability" indicated by an "i".

Property 1 (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 2 (ARR–i–completeness) *A set E of ARRs is said to be* i-complete *if:*

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

ARR–d–completeness and ARR–i–completeness express the theoretical capability of a set of ARRs to be sensitive, hence to detect, any inconsistency between the model SM and observations OBS.

Example – Consider the following signature matrix FS_{ex} assumed to refer to a system with model SM_{ex}. It gathers the signatures of the faults, single and multiple, that may affect four components c_1, c_2, c_3, and c_4. These are composed by the Boolean values of three residuals arising from three ARRs ARR_1, ARR_2, and ARR_3 and are given by the columns of the fault matrix. For example, the fault signature of f_{c_1} is the Boolean vector $(1, 0, 1)^T$. The support of each ARR is given by the rows of the submatrix corresponding to single faults. For example, $supp(ARR_1) = \{c_1, c_2, c_3\}$.

Let us consider the set of ARRs $E_{1,2} = \{ARR_1, ARR_2\}$, then it can be noticed from FS_{ex} that it is d–complete. Indeed, in any of the faulty situations implying a violation of SM_{ex}, there is one ARR in $E_{1,2}$ that is 1, i.e. meaning that it is violated by the corresponding observations. One can also notice that $E_{1,2}$ is not i–complete. The existence of ARR_3 means that we may have a situation in which the subset of components $C = \{c_1, c_2, c_4\}$ is inconsistent, i.e. $SM_{ex}(C) \cup OBS \models \perp$. Nevertheless, since $supp(ARR_1) \not\subseteq C$ and $ARR_2 \not\subseteq C$, property 2 is not fulfilled. As a matter of fact, eventhough ARR_3 seems useless for isolating single faults, one can notice from FS_{ex} that, without ARR_3, no multiple fault of the last column can be discriminated from f_{c_3}.

	f_{c_1}	f_{c_2}	f_{c_3}	f_{c_4}	$f_{c_1c_2}$	*all other multiple faults*
ARR_1	1	1	1	0	1	1
ARR_2	0	0	1	1	0	1
ARR_3	1	1	0	1	1	1

Table 1: Example of a fault signature matrix FS_{ex}

ARR–d–completeness and ARR–i–completeness are key to the comparison of the FDI and DX approaches. The main results can be summarized by the following proposition.

Proposition 1 *(Cordier et al., 2004) 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 an R-conflict;*
2. *If E is a d-complete set of ARRs, and if C is an R-conflict for $(SD, COMPS, OBS)$, then there exists $ARR_i \in E$ that is violated by OBS;*
3. *If E is an i-complete set of ARRs, then given an R-conflict C for $(SD, COMPS, OBS)$, there exists $ARR_i \in E$ that is violated by OBS and supp(ARR_i) is included in C.*

The result 1 of proposition 1 is intuitive and can be explained by the fact that the inconsistencies between the model and observations are captured by R-conflicts in the DX approach and by ARRs violated by OBS in the FDI approach. Consequently, the support of an ARR can be defined as a *potential R-conflict*. This concept is also called *possible conflict* in (Pulido and Gonzalez, 2004a).

The results 2 and 3 of proposition 1 refer to fault detectability and fault isolability. The result 2 outlines the ARR–d–completeness property as the condition for *fault detectability*. From the result 3, the ARR–i–completeness property appears as the condition under which a formal equivalence between R-conflicts and ARR supports holds, as stated by the following corollary.

Corollary 1 *If both the SRE and the ARR–i–completeness properties hold, the set of minimal R-conflicts for OBS and the set of minimal supports of ARRs (taken in any i-complete set of ARRs) violated by OBS are identical.*

The detailed proofs of proposition 1 and corollary 1 can be found in (Cordier et al., 2004).

Let us now analyze the results from a more practical point of view, and point at the differences of the two approaches and how their respective advantages can be put together.

2.3.1 Lesson one: redundant ARRs

An important result coming from ARR–i–completeness refers to redundant ARRs. In FDI, it is generally accepted that if ARR_j is obtained from a linear combination of two other ARRs, ARR_{i_1} and ARR_{i_2}, then ARR_j is redundant (unless some considerations about noises and sensitivity to faults come into play). Nevertheless the i-completeness property states that not only the analytical expression of ARR_j must be taken into account but also its support to conclude about the fact that it is redundant. The formal conditions are stated in the proposition below from (Cordier et al., 2004).

Proposition 2 *A given ARR_j is redundant with respect to a set of ARR_is, $i \in I, j \notin I$, where I is a set of integer indexes such that $card(I) \geq 2$, if and only if $\exists I' \subseteq I$ such that :*

1) $\forall OBS$, if all ARR_is, $i \in I'$, are satisfied by OBS, then ARR_j is satisfied by OBS,

2) $supp(ARR_j) \supseteq supp(ARR_i)$, $\forall i \in I'$.

The above proposition can be explained by the fact that if $supp(ARR_j)$ does not satisfy condition 2, then it captures an inconsistency that is not captured by the initial ARR_is, $i \in I$. It is hence apt, added to the initial ARR_is, to contribute to the achievement of ARR–i–completeness.

2.3.2 Lesson two: exoneration assumptions

The exoneration assumptions, *ARR-exoneration* and *component-exoneration*, used by DX and FDI, respectively, are different.

Definition 8 (ARR-exoneration) *Given OBS, any component in the support of an ARR satisfied by OBS is exonerated, i.e. considered as normal.*

Definition 9 (Component-exoneration) *Given OBS and $c \in COMP$, if $SM(c) \cup OBS$ is consistent, then c is exonerated, i.e. considered as normal.*

The FDI approach generally uses the ARR-exoneration assumption without formulating it explicitly. On the other hand, the DX approach generally proceeds with no exoneration assumption at all. When this is not the case, it uses the component-exoneration assumption and represents it explicitly. If a component c is exonerated, its model is written as:

$$COMP(c) \wedge \neg AB(c) \Longleftrightarrow SM(c)$$

where the simple logical implication, found in (2) for instance, is replaced by a double implication. Explicit assumptions guarantee logical correctness of the DX diagnoses obtained by the DX method. Interestingly, *ARR-exoneration* cannot be expressed in the DX formalism and conversely, *component-exoneration* cannot be expressed in the FDI formalism.

It has been shown that under the same assumptions, in particular in the case of no exoneration, the diagnoses that are obtained by the DX and the FDI approach are the same.

Theorem 1 *(Cordier et al., 2004) Under the i-completeness and no exoneration assumptions, the diagnoses obtained by the FDI approach are identical to the (non empty) diagnoses obtained by the DX approach.*

2.3.3 Lesson three: single and multiple faults

In the FDI approach, because the fault signatures are determined off-line for every fault, the number of considered faults is generally limited. Most of the time, only single faults are considered. On the contrary, the DX approach naturally deals with multiple faults. A consequence is that the number of diagnoses is exponential and this is why it is common to introduce preference criteria, like fault probabilities, to order the diagnoses. Several search methods have been proposed to find the *preferred diagnoses* or to retrieve the diagnoses in preference order (see for instance (Williams and Ragno, 2003; Sachenbacher and Williams, 2004)).

2.3.4 Lesson four: off-line versus on-line

In the FDI approach, ARRs are determined off-line and only a simple consistency check is performed on-line. This may be quite relevant for real-time applications with hard temporal constraints. Inversely, in the DX approach, the whole diagnosis process is on-line, the advantage being that only the models need to be updated in case of any evolution of the system. The two approaches have been integrated to obtain the advantages of both: some DX works have used the idea of the FDI community to construct ARRs off-line (Katsillis and Chantler, 1997; Loiez and Taillibert, 1997; Washio et al., 1999; Pulido and Gonzalez, 2004b) and some FDI works have proposed to base the fault isolation phase on the conflicts derived from violated ARRs (Vento et al., 2010).

3 Bridging the FDI and DX approaches

Building on lessons learned, several works have integrated FDI and DX methods for providing new diagnostic solutions. This section illustrates these integrations with two specific lines, which are the most representative of the emerging synergies.

3.1 Causal model-based diagnosis

In the 90s, the synergies between the FDI and DX approaches found fulfillment in a series of studies proposing *causal models* for diagnostic reasoning. These are perfect examples of integration of the FDI and the DX approaches, the FDI approach being used for fault detection and the DX approach for fault isolation. At that time, causal models were intensively studied by the

Qualitative Reasoning community (Dague and Travé-Massuyès, 2004; Travé-Massuyès and Dague, 2003; Weld and De Kleer, 1989). Whereas (Venkatasubramanian et al., 2003a) overviews how causal models can be used to represent the qualitative behavior of a system and the various ways in which qualitative models can support diagnosis reasoning, this section focuses on another subset of works that enrich the causal structure with quantitative models, hence providing a framework to integrate FDI and DX concepts.

Causal models are based on a graph structure expressing explicitly the dependencies between variables by edges called *influences*. This structure is called a *Causal Graph* from which explanations for the values – normal or abnormal – of variables can be exhibited. Accounting for the dependencies between variables is the basic principle of many diagnostic algorithms. Whereas the standard diagnosis algorithms of DX, such as GDE (De Kleer and Williams, 1987), use the technique known as "dependency recording" (implemented with an ATMS for example) that records dependencies during the inferences, the dependencies used by causal models are exhibited from the outset. These are obtained either directly from expert knowledge as in (Leyval et al., 1994; Gentil et al., 2004), by techniques of causal ordering of the QR community as in (Travé-Massuyès et al., 2001), or from bond-graph models as in (Mosterman and Biswas, 1999; Feenstra et al., 2001).

Causal diagnosis uses the causal graph as an abstract parameterized representation of the system in which every influence is associated with a physical component of $COMP$. Early work was limited to labeling causal influences by the signs giving the direction of change of the influenced variable in relation to the cause variable, thus representing a *Signed Oriented Graph* (Oyeleye and Kramer, 1988; Kramer and Palowitch, 1987). Subsequently, the labeling of influences became more sophisticated and local quantitative models as used in FDI were attached (Leyval et al., 1994; Gentil et al., 2004; Travé-Massuyès et al., 2001). Then, FDI standard techniques for evaluating the residuals could be used for fault detection in the form of a *local* procedure attached to every measured variable of the causal graph. More precisely, fault detection is implemented as an (online) procedure that determines, at each time t and for each measured variable y, the consistency of the measured value $\bar{y}_t$ and the predicted value $\hat{y}_t$ obtained with the – causal – behavioral model of the system. Checking consistency comes back to evaluate the residual:

$$r_t^y = \bar{y}_t - \hat{y}_t$$

If $r_t^y \neq 0$, then y is concluded to be misbehaving, noted MISB(y), and this triggers a diagnostic session.

Diagnostic reasoning is a *global* procedure underpinned by the causal

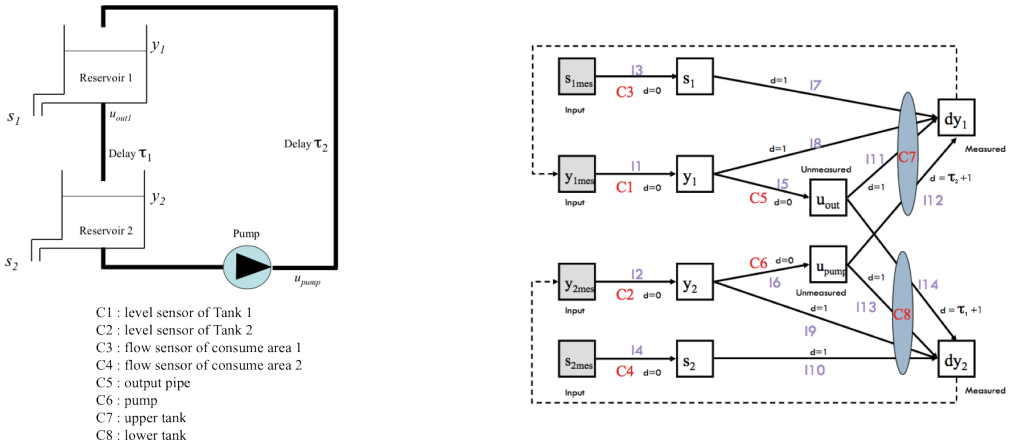

Figure 1: (left) Reservoir system (right) Causal graph with delay labels (from (Travé-Massuyès and Calderon-Espinoza, 2007))

structure labelled by an abstraction of the local models, such as delay times as in Mosterman and Biswas (1999); Travé-Massuyès and Calderon-Espinoza (2007). The detection of a misbehaving variable is interpreted as the violation of one of the influences involved in the prediction of the variable, i.e. of one of the upstream influences in the causal graph. Each influence being associated with a component, the set of upstream influences can be identified as an R-conflict as defined in section 2.2 (or as the support of the ARR that could be built from the local models by performing the variable eliminations indicated by the causal chains). In (Travé-Massuyès and Calderon-Espinoza, 2007), the influences have a *delay* attribute corresponding to the time required for a disturbance acting on the source variable to propagate to the effect variable. Figure 1, right side, provides the causal graph with delay labels for the reservoir system, which involves eight components $Ci, i = 1, \ldots, 8$, shown on the left side. Influences are denoted I_1 to I_{14}. With this information, the elements of the generated R-conflicts can be labeled by a temporal label indicating the latest time at which a component can be considered faulty, i.e. Ci_τ indicates that the component Ci may be faulty from at least $t-\tau$. Diagnoses can be obtained from the R-conflicts by an incremental HS-tree algorithm that generates the hitting sets, according to (Reiter, 1987), while managing the temporal labels (Travé-Massuyès and Calderon-Espinoza, 2007). Figure 2 provides the HS-tree corresponding to the occurrence of two R-conflicts, $\mathscr{C}(dy_2)$ and $\mathscr{C}(dy_1)$, indicated by MISB(dy_2) at time t and MISB(dy_1) at time $t + \delta t$, respectively :

$$\mathscr{C}(dy_2) = \{C1_{\tau_1+1}\ ,\ C2_1\ ,\ C4_1\ ,\ C5_{\tau_1+1}\ ,\ C6_1\ ,\ C8_0\}$$

$$\mathscr{C}(dy_1) = \{C1_1\ ,\ C2_{\tau_2+1}\ ,\ C3_1\ ,\ C5_1\ ,\ C6_{\tau_2+1}\ ,\ C7_0\}$$

The nodes of the HS-tree but leaves are labelled by a conflict set. For each element s in the conflict label of node n, an edge labelled s joins n to a successor node. $H(n)$ is defined as the set of edge labels on the path from n to the root node. The HS-tree is built by considering every conflict in chronological order. Every new conflict is compared to every leaf of the HS-tree, and some new leaves are built if necessary. The resulting HS-tree is pruned for redundant or subsumed leaves before the next conflict is considered. At the end of the procedure, the minimal hitting sets, and hence the minimal diagnoses that explain the detected misbehaving variables, are given by the set of edge labels $H(l)$ associated to the open leaves l of the HS-tree.

In our example, $\mathscr{C}(dy_2)$ is first developed from the Root node at time t. When the second conflict $\mathscr{C}(dy_1)$ is detected, at time $t + \delta t$, the temporal labels of the edge labels from the leaves to the Root are first updated by δt. Then, $\mathscr{C}(dy_1)$ is developed from nodes 3 and 6, given that for $n \in \{1, 2, 4, 5\}$, $H(n) \cap \mathscr{C}(dy_1) \neq \emptyset$. The open leaves of the HS-tree determine the timed diagnoses given in Table 2.

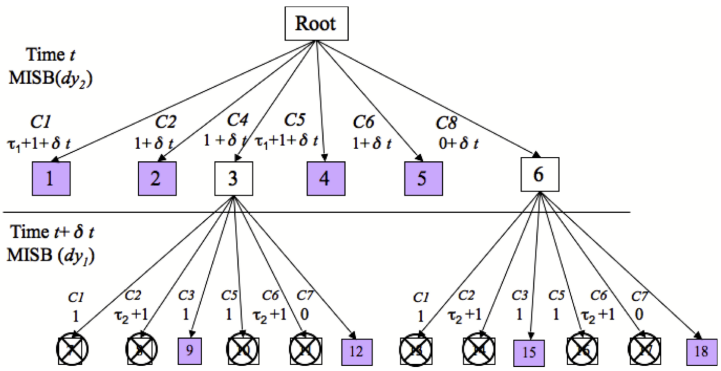

Figure 2: HS-tree with management of temporal labels (from (Travé-Massuyès and Calderon-Espinoza, 2007))

3.2 Model-based diagnosis of hybrid systems

Hybrid systems (Zaytoon et al., 2001; McIlraith et al., 2000; Henzinger, 1996; Lunze and , Eds.) address the need to capture phenomema at different time scales in the same model. They are more particularly devoted to represent systems that have double continuous and discrete dynamics. Forming a specific class of hybrid systems, switched systems undergo abrupt changes of

Table 2: Timed diagnoses updated in time (from (Travé-Massuyès and Calderon-Espinoza, 2007))

Diagnoses at time t	Diagnoses at time $t + \delta t$
$D1 = \{C1_{\tau_1+1}\}$	$D1 = \{C1_{\tau_1+1+\delta t}\}$
$D2 = \{C2_1\}$	$D2 = \{C2_{\tau_2+1}\}$
$D3 = \{C4_1\}$	$D3 = \{C5_{\tau_1+1+\delta t}\}$
$D4 = \{C5_{\tau_1+1}\}$	$D4 = \{C6_{\tau_2+1}\}$
$D5 = \{C6_1\}$	$D5 = \{C4_{1+\delta t}, C3_1\}$
$D6 = \{C8_0\}$	$D6 = \{C4_{1+\delta t}, C7_0\}$
	$D7 = \{C8_{0+\delta t}, C3_1\}$
	$D8 = \{C8_{0+\delta t}, C7_0\}$

dynamics upon switches. They are particularly relevant to diagnosis because switching dynamics is just representative of fault occurrence. Switched systems may be modeled by a state transition system whose states represent the modes of operation – normal or faulty – for which the continuous dynamics are specified. The modeling of such systems and the associated diagnostic algorithms make use of continuous and discrete formalisms, defining an ideal field for the integration of FDI and DX methods. In this case, the emphasis is on coupling the search methods used by DX with FDI methods in the form of state estimators, parameter estimators or sets of ARRs. Search methods allow one to deal with the combinatorial explosion of the number of trajectories to be tracked. FDI methods can also be coupled with discrete event system methods that are investigated in both the FDI and DX communities.

In the DX community, researchers dealing with hybrid systems first tried to abstract the continuous dynamics in qualitative terms. The Livingstone diagnostic engine Pell et al. (1998); Williams and Nayak (1996), from NASA, that flew on board the DS-1 probe, was among the first to be qualified as hybrid. This engine was rooted in the DX approach, with a model written in propositional logic, and behavioral equations reflecting the continuous dynamics in the form of logical relationships expressing qualitative constraints. The qualitative abstraction required *monitors* between the sensors and the model to interpret continuous signals in terms of discrete modalities. This approach suffered from two drawbacks : the thresholds were extremely difficult to determine, and detection sensitivity was quite poor. Subsequent research thus replaced the qualitative abstractions by classical differential equations, resulting in real hybrid models interlinking continuous and discrete states.

Although the different classes of hybrid models show differences in their

formalization, a hybrid model may be exemplified by the following tuple, which makes explicit continuous and discrete dynamics as well as their interaction (Bayoudh et al., 2008a):

$$S = (\zeta, Q, \Sigma, T, C, (q_0, \zeta_0)) \tag{3}$$

where:

- ζ is the set of continuous variables including state variables, input/output variables, and possibly noises, which are functions of time t. Some are measured, others not.

- Q is the set of discrete system states. Each state $q_i \in Q$ represents a mode of operation of the system.

- $\Sigma = \Sigma_s \cup \Sigma_c \cup \Sigma_f$ is the set of events including spontaneous mode switching events, discrete control inputs, and fault occurrences. Events corresponding to spontaneous mode changes are issued upon guards that depend on continuous variables, as defined below. Σ can be partitioned as $\Sigma_o \cup \Sigma_{uo}$ where Σ_o represents the set of observable events and Σ_{uo} represents the set of unobservable events. $\Sigma_f \subseteq \Sigma_{uo}$, $\Sigma_c \subseteq \Sigma_o$ and Σ_s may have elements in both sets.

- $T \subseteq Q \times \Sigma \rightarrow Q$ is the partial transition function. A transition $t(q_i, \sigma_{ij}, q_j)$ may be guarded by a condition given as a set of equations $\mathcal{G}(t(q_i, \sigma_{ij}, q_j)) = g_{ij}(x, \theta_g) = 0$, θ_g being a constant parameter vector. Then σ_{ij} results from the state $x(t)$ hitting the guard g_{ij} at some time instant t^* and is not observable. A reset map $\mathcal{R}_{ij}$, possibly equal to the identity, is specified.

- $C = \bigcup_i C_i$ is the set of system constraints linking continuous variables. C_i denotes the set of constraints associated to mode q_i, which are given in state-space form by the following continuous time state-evolution and output equations:

$$\begin{cases} \dot{x}(t) &= f_i(x(t), u(t), \epsilon(t)) \\ y(t) &= g_i(x(t), u(t), \epsilon(t)) \end{cases} \tag{4}$$

  where $u \in \mathbb{R}^{n_u}$, $x \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^{n_y}$ are the input, output, state vectors, respectively, and $\epsilon \in \mathbb{R}^{n_\epsilon}$ denotes some noise vector. The variables gathered in these vectors belong to ζ.

- $(\zeta_0, q_0) \in \zeta \times Q$ is the initial condition of the hybrid system.

In the hybrid state (ζ, Q), only the discrete state Q is representative of the operation mode of the system and estimating Q is hence the target of diagnosis. Because the evolution of discrete states is closely tied to the evolution of the continuous states, the diagnosis problem is often brought back to the problem of estimating the full hybrid state.

In theory, hybrid estimation must consider all the possible mode sequences with the associated continuous evolutions, which is exponentially complex. Consequently, many sub-optimal methods have been proposed, of which we can distinguish the following three classes:

- methods known as *multiple-model estimation*, rather rooted in the FDI field (Ackerson and Fu, 1970; Blom and Bar-Shalom, 1988; Li and Bar-Shalom, 1996),
- *particle filtering* methods (de Freitas, 2002; Verma et al., 2004; Narasimhan et al., 2004a), which are found both in FDI and DX,
- methods that address hybrid aspects in a *dedicated* manner (Hofbaur and Williams, 2002b; Benazera et al., 2002; Narasimhan and Biswas, 2002; Benazera and Travé-Massuyès, 2009), taking advantage of FDI and DX approaches.

3.2.1 Multiple-model estimation methods

Multiple-model estimation methods, inspired by the IMM algorithm of (Blom and Bar-Shalom, 1988), are formulated in a probabilistic framework. IMM considers the problem of estimating the state of a system with Markovian switching parameters:

$$\begin{cases} x(t) & = a(\theta_t)x(t-1) + b(\theta_t)\epsilon_1(t) \\ y(t) & = h(\theta_t)x(t) + g(\theta_t)\epsilon_2(t)) \end{cases} \tag{5}$$

where θ_t is a finite state Markov chain taking a finite number of N values according to a transition probability matrix, and $\epsilon_1(t)$ and $\epsilon_2(t)$ are mutually independent white Gaussian processes.

IMM avoids the exponential growth of the number of Gaussian hypotheses by merging the hypotheses. Continuous estimates arising from N Kalman filters are merged according to a likelihood measure to provide a *belief state* in the form of a probability distribution over states at the current time. The likelihood measure is provided by every filter and expresses the degree of consistency between an hypothesis and observations but it takes also into account the information carried by the probabilities of transitions. The timing for merging is studied in (Blom and Bar-Shalom, 1988), which provides

an efficient procedure. In (Washington, 2000), which suggests combining a process of continuous state estimation given by Kalman filters with an estimation process of the discrete state based on a POMDP (Partially Observed Markov Decision Process), the estimation method is successfully applied to fault identification on rovers.

(Hofbaur and Williams, 2002b) also unifies continuous state estimators with a belief calculated from Hidden Markov Models (HMM). The approach is quite similar to the one of (Washington, 2000) in that it uses a set of Kalman filters, but it adopts an aggressive strategy for adjusting the number of tracked trajectories to the computing resources available. This strategy uses an A^* algorithm, which selects the most likely state trajectory branches and takes into account unknown fault modes. It is implemented in the hME engine by (Hofbaur and Williams, 2002a, 2004).

Interestingly, it was shown by (Benazera and Travé-Massuyès, 2009) that considering bounded instead of probabilistic uncertainty for noises and parameters of the continuous processes allows one to merge the hypotheses with the same discrete state estimate by computing the convex hull of continuous estimates. This kind of merging, only possible in a set-membership framework (Milanese and Vicino, 1991; Kieffer and Walter, 2011) was implemented by interval analysis methods in (Benazera and Travé-Massuyès, 2009) and later in (Ramdani and Nedialkov, 2011) and (Maiga et al., 2013). It controls effectively the number of tracked hypotheses and may avoid truncating/approximating the set of hypotheses.

3.2.2 Particle filtering methods

Particle filtering methods are based on simulation and a Bayesian update of prior beliefs. They proceed by sampling the initial probability distribution. The underlying principle is the approximation of the conditional state probability distribution $p(x_{0:t} \mid y_{0:t})$ by a swarm of points called *particles*. These particles sample the state-space and carry a weight representing a discrete probability masse. Particles are generated and recursively updated given a nonlinear process model, which describes the evolution in time of the system under analysis, and a measurement model:

$$\begin{cases} x(t) &= f_t(x(t-1), w(t)) \\ y(t) &= h_t(x(t), v(t)) \end{cases} \tag{6}$$

where $\{w(t)\}_{t\geq 0}$ and $\{v(t)\}_{t\geq 0}$ are sequences of random variables, not necessarily Gaussian, as well as a set of available measurements $Y = \{Y_t, t \in \mathbb{N}\}$ and an initial estimation for the state probability density function $p(x(0))$.

With enough samples, particle filtering approaches the optimal Bayesian estimate while presenting the advantage to accept probability distributions of any kind. We refer the reader to (Arulampalam et al., 2002) for a tutorial on particle filtering methods. Despite their advantage, it should be noted that they are difficult to apply to the diagnosis problem since the probabilities of faults are generally very low compared to those of nominal states. The number of particles needed to track the faulty states is hence very high. Different strategies were tested to provide solutions to this problem (Verma et al., 2003; Thrun et al., 2002). Among them, (Narasimhan et al., 2004b) combines particle filtering and the DX based diagnosis approach Livingston L3 Williams and Nayak (1996); Kurien and Nayak (2000). In this approach Rao-Blackwellized Particle Filtering is used to track the nominal behavior, an n-step prediction scheme is used for fault detection and L3 is used to generate a set of candidates that are consistent with the discrepant observations. This candidates are then tracked by the particle filtering scheme.

3.2.3 Hybrid system diagnosis dedicated methods

As we have seen before, the combinatorial explosion inherent to hybrid estimation leads to solutions providing an approximated belief state. All previous methods suffer from the problem of *hypothesis loss*, which may be critical for a diagnosis application. The set of modes estimated over time, i.e. the diagnoses, takes the form of a set Γ of trajectories generated by a decision process that abandons the trajectories whose belief is below a given threshold. Unfortunately, Γ is reused in the calculation of future estimates and it may happen that the actual trajectory of the system is no longer in Γ, thus producing an erroneous diagnosis. Several studies have addressed this problem.

(Nayak and Kurien, 2000) proposes a solution based on a *a posteriori* progressive exploration of the space guided by fault *ranks*. However, this requires considering a large time window backwards from the current time, which may even include the initial time. In this case, the current state must be re-predicted from the whole system's history. The diagnostic engine KOALA (Benazera and Travé-Massuyès, 2003) proposes a variant of the previous algorithm with a *revision* procedure that produces diagnoses from the current trajectory by performing minimal partial changes.

There have been other attempts to remedy the loss of solutions. In (Lerner et al., 2000) (Lerner et al., 2002), the authors reduce the set of tracked trajectories by aggregating those using similar assumptions. In addition, they develop a smoothing strategy that propagates the current beliefs back in time to improve the prediction of system states.

Recently, an alternative approach to hybrid state estimation, called rME, was proposed based on the hybrid model (3). rME focuses on mode estimation, the actual diagnosis, and ignores continuous state estimation (Bayoudh et al., 2008c). rME is based on the parity space approach as presented in section 2.1 for processing continuous data. A set of ARRs is determined for each mode of the hybrid system from the associated continuous model. They are all put together in a vector. When evaluated with the measurements in a given mode, this vector defines its *mode signature*. The mode signature changes that result from transitioning from one mode to another are abstracted as a set of events, called *signature-events*, by the abstraction function f_{Sig}:

$$\begin{array}{rcl} f_{Sig} : Q \times T(Q, \Sigma) & \longrightarrow & \Sigma^{Sig} \\ (q_i, q_j) & \longmapsto & \delta_{ij} \end{array} \tag{7}$$

The event δ_{ij} is observable and noted Ro_{ij} if the mode signature of the source mode q_i is different from the mode signature of the destination mode q_j, i.e. $Sig(q_i) \neq Sig(q_j)$. δ_{ij} is unobservable and noted Ruo_{ij} otherwise [4]. Hence Σ^{Sig} is partitionned in a set of observable signature-events Σ_o^{Sig} and a set of unobservable signature-events Σ_{uo}^{Sig}.

The set $\Sigma^{Sig} = \Sigma_o^{Sig} \cup \Sigma_{uo}^{Sig}$ is used to enrich the underlying DES $M = (Q, \Sigma, T, q_0)$, producing an automaton that captures both the abstraction of the continuous dynamics and the discrete dynamics. A classical DES diagnosis approach, like the diagnoser approach (Sampath et al., 1995), then permits to track the current mode of the system based on the received observable events, i.e. to provide the diagnosis.

Lately, the mode estimation rME approach (Bayoudh et al., 2008c) was combined with the state estimation approach hME (Hofbaur and Williams, 2004), resulting in mutual enrichment (Rienmuller et al., 2009; Rienmüller et al., 2013). hME brings an estimate of the continuous state that is not provided by rME. In addition, among all modes corresponding to the current hypotheses of hME, rME can identify those that are consistent with observations, providing a reduced set of possible modes to hME and significantly reducing the number of hypotheses to be tracked by the hybrid estimator.

[4] By construction, mode signatures cannot change while being in the same mode.

4 Bridges for learning diagnosis models : an example with chronicle learning from time series

The main weakness of MBD diagnosis approaches, either from FDI or from DX, is to require well-formalized models that must be built from the available knowledge about the system. However, knowledge may be scarce. On the other hand, nowadays systems generate big amounts of data that are stored. Data-driven diagnosis approaches take advantage of these data and use a whole spectrum of machine learning methods to directly assess the state of a system and provide a diagnosis. An alternative way is to use machine learning to generate models automatically, hence bridging with MBD diagnosis methods. Regression analysis is well-known for learning continuous models from measured time series but learning diagnosis-oriented discrete event models requires bridges with symbolic learning rooted in AI. This is a promising research track that can produce methods benefiting from all the available information in the form of data and knowledge.

To exemplify this track, let us consider the problem of generating discrete-event models automatically from raw data in the form of time series. Some works have built on classifiers to generate automata or Markov chain models, in which states represent operation modes of the system (Kempowsky et al., 2006a; Omlin et al., 1998; Kempowsky et al., 2006b). In this paper, we illustrate the generation of chronicles as used in the AI community (Dousson and Duong, 1999; Cram et al., 2011).

We are interested in learning high level temporal patterns that capture the behavior of dynamic processes at an abstract level based on events. In the diagnosis context, these patterns can be used as signatures of specific situations and associated to decision rules specifying which actions must be taken in different situations (Subias et al., 2010).

4.1 Event-based abstraction

The system is assumed to be described through a set of *features* whose values change over time with the system's evolution. The data samples are given in terms of the set of features $\{\chi^1, \ldots, \chi^n\}$. Every χ^j takes its value in the set $\mathcal{U}^j$, called the *domain* of χ^j. Therefore any sample can be represented by a vector $\vec{x} = (x^1, \ldots, x^n)^T$ of $\mathcal{U} = \mathcal{U}^1 \times \cdots \times \mathcal{U}^n$, so that every component x^j corresponds to the feature value χ^j qualifying the object $\vec{x}$. The subset of $\mathcal{U}$ formed by these vectors is called the *data base*.

Samples are indexed by time, a sample taken at time t_i is represented

by a vector $\vec{x}_{t_i} = (x^1_{t_i}, \ldots, x^n_{t_i})^T$. The value taken by the feature χ^j across time can be considered as a random variable $x^j_t, t \in \mathbb{Z}$. Then the time series corresponding to the samples, taken from time t_i to time t_f, is noted $\boldsymbol{X}_{t_i - t_f} = \{\vec{x}_t, t = t_i, \ldots, t_f\} = \langle \vec{x}_{t_i}, \ldots, \vec{x}_{t_f} \rangle$.

Events are a symbolic representation of time-stamped specific singularities of the signals represented by the time series. The concept of *event type* expresses a change in the value of a given domain feature or set of features. Let us define E as the set of all event types and define the concept of *event*.

Definition 10 (Event) *An* event *is defined as a pair* (e_i, t_i)*, where* $e_i \in \Sigma$ *is an event type and* t_i *is an integer called the event date.*

Time representation relies on the time point algebra and time is considered as a linearly ordered discrete set of instants whose resolution is sufficient to capture the system's dynamics (Vilain and Kautz, 1986).

Definition 11 (Temporal sequence) *A* temporal sequence *on* Σ *is an ordered set of events denoted* $\mathcal{S} = \langle (e_i, t_i) \rangle_{i \in \mathbb{N}_l} = \{(e_i, t_i)\}_{i \in \mathbb{N}_l}$ *such that* $e_i \in \Sigma, i = 1, \ldots, l$ *and* $t_i < t_{i+1}, i = 1, \ldots, l-1$*, where* l *is the dimension of the temporal sequence* $\mathcal{S}$*.*

The time series $\boldsymbol{X}_{t_i - t_f}$ for a set of features can be abstracted in terms of a temporal sequence. This abstraction can be achieved as proposed in (Aguilar-Castro et al., 2012) by applying clustering methods to the set of samples in $\boldsymbol{X}_{t_i - t_f}$. The samples are thus clustered into a set of classes. The conditions that underly the transitions between classes can then be used to define event types and the transition dates provide the event dates.

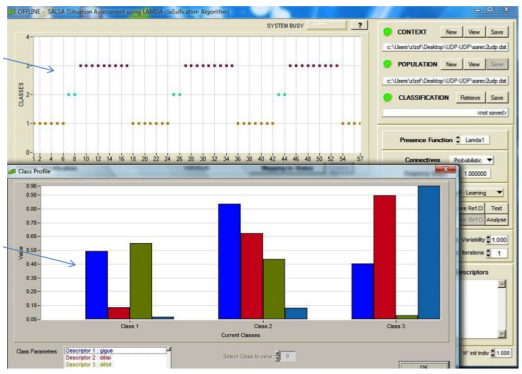

Figure 3: Event-based abstraction with LAMDA

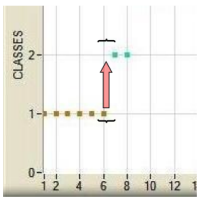

Figure 4: Generation of events

The event-based abstraction is illustrated by figure 3 in which the fuzzy clustering method LAMDA (Aguilar-Martin and Lopez de Mantaras, 1999; Kempowsky et al., 2003) has been used. Three time series corresponding to the same dynamic situation have been concatenated and given to LAMDA, hence the repetition of the same clustering pattern in three classes appearing in the top of the figure. The bottom bar diagrams provide the so-called *profile* of the three classes. Figure 4 illustrates that an event is generated when there is a class transition.

4.2 Chronicle concept

Chronicles have been introduced in AI as a way to express temporal information about a domain (Dousson et al., 1993). Chronicles are a rich formalism allowing one to describe observable event patterns corresponding to the behaviors one wants to detect. Most importantly, chronicles come together with powerful event recognition engines that are able to recognize them on the fly given a temporal sequence as input (Artikis et al., 2010). With chronicles one may want to represent dynamic situations like :

- an event type e_i is immediately followed by event type e_j,
- an event type e_i is followed by event types e_j and e_k after less than 10 time units,
- an event type e_i occurs after 8 occurrences of an event type e_j.

To represent this type of domain evolutions, we consider that event dates may be constrained. A time interval I is expressed as a pair $I = [t^-, t^+]$ corresponding to the lower and upper bound on the temporal distance between two time points t_i and t_j, $i \leq j$. Given two event dates t_i and t_j, we express temporal constraints of the form $t_j - t_i \in [t^-, t^+]$. Consider two events (e_i, t_j) and (e_j, t_j), then if their dates t_i and t_j satisfy the temporal

constraint $t_j - t_i \in [t^-, t^+]$, we write $\tau_{ij} = e_i[t^-, t^+]e_j$ and say that the events are temporally constrained by τ_{ij}.

Definition 12 (Chronicle) *A* chronicle *is a pair* $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ *such that* $\mathcal{E} \subseteq \Sigma$ *and* $\mathcal{T} = \{\tau_{ij}\}_{1 \leqslant i < j < |\mathcal{E}|}$. $\mathcal{E}$ *is called the* typology *of the chronicle and* $\mathcal{T}$ *is the set of* temporal constraints *of the chronicle.*

A chronicle $\mathcal{C}$ represents an evolution pattern involving a subset of events $\mathcal{E}$ and a set of temporal constraints $\mathcal{T}$ linking event dates. Chronicles are a special type of temporal pattern, where the temporal order of events is quantified with numerical bounds and reflects the represented piece of temporal evolution. Figure 5 is an example of chronicle in which $\mathcal{E} = \{e_1, e_2, e_3\}$ and $\mathcal{T} = \{\tau_{13} = e_1[3, 6]e_3, \tau_{23} = e_2[3, 9]e_3\}$.

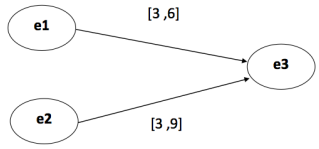

Figure 5: Example of chronicle

The occurrences of a given chronicle $\mathcal{C}$ in a temporal sequence $\mathcal{S}$ are denoted by subsequences called chronicle *instances*.

Definition 13 (Chronicle instance) *An instance of* $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ *in a temporal sequence* $\mathcal{S}$ *is a subset of events* $\mathcal{E}'$ *of* $\mathcal{S}$ *such that* $\mathcal{E}'$ *is isomorphic to* $\mathcal{E}$.

The above definition means that $|\mathcal{E}'| = |\mathcal{E}|$ and that the events of $\mathcal{E}'$ satisfy all temporal constraints $\mathcal{T}$ of the chronicle $\mathcal{C}$.

4.3 Learning chronicles

Let us consider a temporal sequence $\mathcal{S}$ corresponding to a given dynamic situation. The goal is to learn the chronicle(s) that best characterize this situation from $\mathcal{S}$. A commonly used approach to elaborate a solution for this problem relies on the notion of *frequency*.

Definition 14 (Frequency of a chronicle in a temporal sequence) *The* frequency *of a chronicle* $\mathcal{C}$ *in a temporal sequence* $\mathcal{S}$, *noted* $f(\mathcal{C}|\mathcal{S})$, *is the number of instances of* $\mathcal{C}$ *in* $\mathcal{S}$.

It is understood that most frequent chronicles are those that supposedly best represent the situation. On the other hand, given a set of event types Σ, the space of possible chronicles can be structured by a *generality relation*.

Definition 15 (Generality relation among chronicles) *A chronicle* $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ *is* more general than *a chronicle* $\mathcal{C}' = (\mathcal{E}', \mathcal{T}')$*, denoted* $\mathcal{C} \sqsubseteq \mathcal{C}'$*, if* $\mathcal{E} \subseteq \mathcal{E}'$ *or* $\forall \tau_{ij} \in \mathcal{T}, \tau_{ij} \supseteq \tau'_{ij}$.

Having defined the above concepts, the problem can be formulated as: given a temporal sequence $\mathcal{S}$, find all less general chronicles $\mathcal{C}$ in $\mathcal{S}$ such that $f(\mathcal{C}|\mathcal{S}) \geq f_{th}$, where f_{th} is a minimum frequency threshold. Symbolic learning methods like those proposed in (Cram et al., 2011) can be used to solve this problem.

Chronicle mining is illustrated by figures 6 and 7. Figure 6 provides an example of three event sequences obtained from three time series corresponding to the same situation. Figure 7 gives the less general chronicle that characterizes the three sequences.

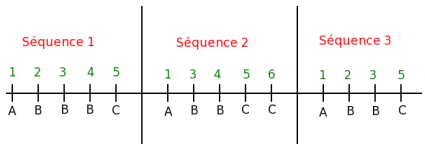

Figure 6: Example of three event sequences obtained from time series of the same situation

```
message A
message B
message C

chronicle Chronicle1
{
        event(A,t1)
        event(B,t2)
        event(C,t3)

        t2-t1 in [2.0,2.0]
        t3-t1 in [4.0,4.0]
        t3-t2 in [2.0,3.0]
}
```

Figure 7: Chronicle characterizing the three sequences

5 Diagnosis in fault management architectures

In the last few years, accounting for fault analysis and diagnosis in the solutions produced for autonomous architectures on one side, and for maintenance architectures on the other side has been of concern. Diagnosis is thus seen as a piece of the puzzle that constitutes the whole fault management architecture. This section discusses the openings that exist and the challenges for future research bridges in these areas.

5.1 Autonomous architectures

During many years, the focus of the work dealing with autonomy has been put on planning and execution and a proper well-identified diagnosis module was absent from the proposed architectures (Alami et al., 1998; Finzi et al., 2004). However, it is now more and more recognized that control and planning must cooperate with diagnosis so as to jointly solve the state tracking and decision problems as illustrated in the full template architecture of figure 8. The system is submitted to the controller actions, which are taken according to the plan decided by the *Mission Planner*, the current diagnoses provided by the *Diagnoser* and the plan elaborated by the *Diagnosis Planner* to disambiguate the diagnoses, i.e. to achieve *active diagnosis*. Two knowledge bases support the above mentioned tasks: the knowledge constituting the behavioral model of the system and the knowledge about the system mission. Examples of autonomous architectures that include a diagnoser can be found in (Williams et al., 2003; Verfaillie and Charmeau, 2006). Although the presence of a diagnosis planner is still uncommon, examples can be found in (Bayoudh et al., 2009; Chanthery et al., 2010).

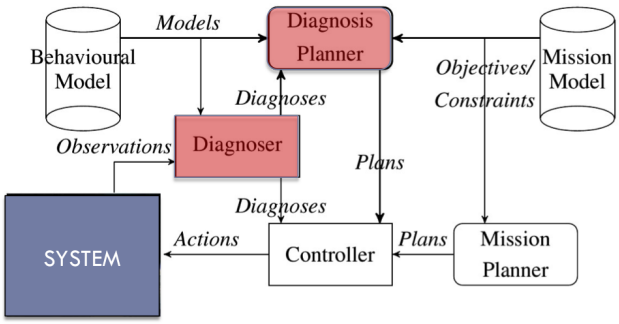

Figure 8: Autonomy template architecture

In this framework, the problem of fault tolerant control that refers to

the interactions between the *Diagnoser* and the low level control, has been intensively studied by the Control community (Blanke et al., 2003). Tools for deciding about system reconfigurability have been proposed as well as solutions to reconfigure control laws on line in order to guarantee the stability of the system and to keep the state of the system close to the desired state even when a fault occurs.

Nowadays challenges involve bridging diagnosis with high level planning. In the following, two types of interactions that involve the Diagnoser are analysed, those for which diagnosis serves planning, known as *plan diagnosis*, and those for which planning acts for improving diagnosis, known as *active diagnosis*.

5.1.1 Diagnosis for planning : plan diagnosis

Planning generates a control program, also known as a *plan*, which describes the sequence of actions necessary to achieve some pre-defined goals. Traditionally, the planning community works with the assumption of full observability (Ghallab et al., 2004). In practice, this assumption must be relaxed to consider partial observability, a situation that is more likely to be encountered in reality and where having diagnosis capabilities becomes essential (Benazera, 2009).

Diagnosis serves in determining which action(s) failed and what went wrong in terms of the internal states of the system so that the system can possibly recover from the fault(s). (De Jonge et al., 2006) introduced two general types of plan diagnosis: *primary plan diagnosis* identifying the incorrect or failed execution of actions, and *secondary plan diagnosis* that identifies the underlying causes of the faulty actions. This two types were later extended with *agent diagnosis* by (Micalizio and Torasso, 2007) because the problem of plan diagnosis is commonly formulated within a distributed perspective where each "agent" is responsible for a part of the total plan. In this context, (Witteveen et al., 2005) showed how plan-diagnoses of the partial plans are related to diagnoses of the total plan and how global diagnoses can be obtained in a distributed way. Further work has considered challenging partial observability situations in which available information may not be enough for univocally determining the outcome of the actions that are executed (Micalizio and Torasso, 2009).

In the above works, actions are considered instantaneous. But real-world systems and processes (e.g. motors, actuators) operate dynamically in continuous time, denoted t. To know what is happening on such systems, we often need to make measurements using a sampling time, denoted $t_k, k \in \mathbb{N}$, at a sampling interval T_s. When the behaviour of the system is abstracted for

reasoning, and particularly for planning, both continuous time and sampling time considerations are not of immediate relevance. Rather, we are interested in what is happening at logical time (l), e.g. action execution cycle points (l_{act}) or observation time points (l_{obs}). Time considerations and different time scales are hence essential if one wants to consider the continuous dynamics effects. But considering nominal and fault models as continuous models has major implications on how planning and diagnosis are handled. Time must be introduced into the models of actions, i.e. actions must have a duration, allowing the specification of continuous dynamics in operation for the actions involved. On the other hand, the planning formalism must be able to express and handle temporal constraints. Let us notice that (Micalizio et al., 2011) considers plans with durative actions, although time is not continuous. The objective of this work is to prevent problems rather than repair them : when a temporal behavioral pattern expresses a deviation with respect to the nominal expected behavior, the modality of execution of the current action may be changed. Requirements about time and the problem of unifying diagnosis and planning models are still quite open and call upon novel developments.

5.1.2 Planning for diagnosis : active diagnosis

Active diagnosis is another aspect of the picture in which diagnosis and decision theories can cooperate. Active diagnosis relies on applying specific inputs to the system so that additional symptoms that permit refining the diagnosis are exhibited.

For DES, (Sampath et al., 1998) can be cited as the seminal work. A DES is modeled by an automaton, and active diagnosis is formulated as a supervisory control problem as defined in (Ramadge and Wonham, 1989). The idea of this approach is to design the controller so that specific actions that may drive the system into non diagnosable regions are forbidden. Diagnosability of the system at run time relies on the controller design performed off-line. The system is hence maintained "actively" diagnosable and its diagnoser produces non ambiguous outputs. In other words, active diagnosis is achieved by preventing inappropriate control actions.

(Daigle et al., 2010) argues that active diagnosis requires both prevention and execution of actions and combines the two above perspectives, building on preliminary ideas presented in (Bayoudh et al., 2008b), and those of (Sampath et al., 1998). Active diagnosis is split into two parts, a diagnoser and a controller that aims to compute an appropriate sequence of mode change control actions to best satisfy the control and diagnosis objectives.

The so-called *pervasive diagnosis* paradigm proposed in (Kuhn et al.,

2008) is based on different ideas, while still coupling diagnosis and planning. Because the planner is generally allowed to achieve mission goals in multiple ways, (Kuhn et al., 2008) proposes a heuristic search for *informative plans*, which are the plans that provide best diagnosability. Here, no specific active diagnosis actions are synthesized. Finally, the *safe diagnosability* and *active safe diagnosis* approach defined by (Paoli and Lafortune, 2005) for DES and later extended by (Liu and Qiu, 2008) to stochastic DES aims to achieve fault diagnosis prior to the execution of a given set of forbidden actions in the failed mode of operation of the system, hence preventing faults from developing into failures that could cause safety hazards.

Unlike these two latter works, the approach of (Bayoudh et al., 2009) adopts the perspective that the operation of the system can be suspended, and that it is up to a decision module to decide whether it is best to continue operating the system or to run an active diagnosis session. The sequence of actions to be executed for active diagnosis hence do not interlink with actions dedicated to control. It is also different from (Sampath et al., 1998) in that regions of poor diagnosability are accepted because these are common for many systems in practice. When the system is in an ambiguous diagnosis state, active diagnosis relies on executing actions to drive the system towards more diagnosable regions. The method achieves the best possible diagnosis refinement, although it may happen that diagnosability cannot be improved.

In the hybrid framework, both continuous and discrete control actions can be used, in an interlinked way, to perform active diagnosis. These interlinked actions ultimately act towards putting the system in a goal configuration, i.e. a goal behavioral mode, by driving it through a selected sequence of intermediary operation modes.

In (Bayoudh et al., 2009; Chanthery et al., 2010) active diagnosis is formulated as a conditionnal planning problem. Starting with an ambiguous diagnosis state, a sequence of control actions that sequencially reduce the ambiguity of an initial diagnosis is delivered. To do so, given an initial uncertain state q_{D_1}, an *active diagnoser*[5] is interpreted as an *AND-OR* graph rooted in q_{D_1}. The states with multiple outgoing transitions initiating sequences of consecutive controllable events c_i are interpreted as "OR" nodes (squares in Figure 9), the states whose outgoing transitions are labelled with induced events ic_i, i.e. induced by the continuous dynamics, are interpreted as "AND" nodes (circles in Figure 9). Then, planning algorithms rooted in the AI field can be applied to solve the conditional planning problem,

[5]The active diagnoser is obtained from the diagnoser (Sampath et al., 1995) by constraining the outgoing transitions of a state to those labelled by events (i.e. actions) permitted in all the underlying original automaton states (Bayoudh et al., 2009)

for instance the classical *MINIMAX* algorithm (Russell et al., 1995) or an AO*-type algorithm as proposed in (Chanthery et al., 2010).

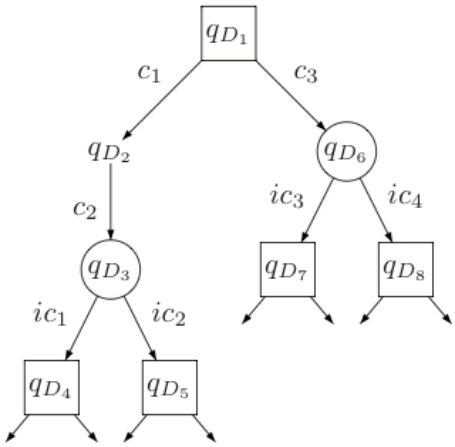

Figure 9: Interpretation of the active diagnoser as an AND-OR graph

Optimal input design as known in the Control field can be seen as a continuous control version of active diagnosis (Mehra, 1974; Jauberthie and Chanthery, 2013).

5.2 Maintenance architectures

In maintenance programs, diagnosis appears as an ingredient for achieving condition monitoring and allows one to make adjusted predictions about the remaining useful life (RUL) of the system and its components. From this information, maintenance actions can be scheduled at best.

Prognostic is often the result of some reliability statistics drawn for an entire population of components of a given category from a set of scenarios implementing different stress conditions (Vachtsevanos et al., 2006; Saxena et al., 2010). The failure time of a system is represented by a random variable X for which the probability density function (pdf) is obtained from statistical prediction analyses. The probability for a system's RUL to be lower than a time t_p is hence given by:

$$P(X \leq t_p) = \int_0^{t_p} f(t)dt, \tag{8}$$

where $f(t)$ represents the failure time pdf. The failure rate of a system, noted $F(t)$, represents the probability for the system to fail in the time interval $]t, t + dt]$ given that it was sane at time t :

$$
\begin{aligned}
F(t)dt &= P(X \leq t + dt | X > t) \\
&= \frac{P(t < X \leq t+dt)}{P(X > t)}.
\end{aligned} \tag{9}
$$

For a specific component in operation, it is possible to assess its current healh status, resulting from the specific stress it undergoes, and this is where diagnosis comes into play. Diagnosis provides a regularly updated health status of specific components that can be used to update the initial state for predicting their RULs (Ribot et al., 2009, 2008; Mohanty et al., 2010; L.Hedjazi et al., 2011). Health status updating prognosis differs from prognosis performed from reliability analyses and it is sometimes called "adaptive prognosis". In most model-based prognosis literature (Orchard and Vachtsevanos, 2009; Daigle and Goebel, 2013; Bregon et al., 2012), it is understood that diagnosis is a necessary part of the architecture, so *adaptive* is generally omitted.

Diagnosis determines the state of the system, and of each of its components, as being consistent with the current observations, then prognosis can determine the state that will be reached in the future and that is consistent with the current diagnosis and with the ageing model of the system. The RUL can then be evaluated for every component in adaptation with the health status reported by diagnosis. Although diagnosis injects precision, the future stress conditions are inherently uncertain, resulting in uncertain RUL estimation. Scheduling maintenance actions from this information hence requires sophisticated algorithms to which the AU community contributes (Pinedo, 2012). Other tools that are generally used in this area are statistics, machine learning and estimation methods.

6 Conclusion

In this paper, different facets of diagnosis investigated in the Control or the AI fields are discussed. Diagnosis benefits from a wide spectrum of methods and each of them brings in different potentialities. Having the whole picture helps understand the different variants of the diagnosis problem in relation with the different application domains. This is also essential to propose suitable and relevant solutions to real diagnosis problems.

This paper does not survey the different diagnosis approaches (see (Venkatasubramanian et al., 2003c,a,b) for a survey of this kind) but it surveys the "bridging" works that integrate approaches from the Control or the AI fields.

The model-based FDI and DX tracks are given special attention because they can be put in correspondence when they seem to have nothing in com-

mon on the surface. Their comparative analysis is quite instructive and the lessons learned from this exercise are pointed out. The different hypotheses adopted by the two communities can be explained by their historical background. The Control community approaches the diagnosis problem from the fault detection point of view, providing numerous efficient methods to cope with stochastic or error bounded noisy and disturbed systems, whereas the AI community is prone to use abstract symbolic models that can support diagnosis reasoning, understood here as fault isolation. This is why hybrid models, that combine continuous and discrete paradigms, open nice perspectives for bridging diagnosis approaches from these two fields.

In section 4, learning the models that support diagnosis reasoning is shown to be a rich field for bridging theories. The machine learning methods that are the foundations of process history diagnosis-based methods can be exploited to build sophisticated models that can then support more sophisticated diagnosis reasoning. This is illustrated with the automatic generation of chronicles in section 4.3. Classification methods are commonly applied to historical data to provide a means to recognize on-line the operation mode of a system. It is shown that it can be used to learn sequences of events instead. Submitting these sequences to a second learning level allows one to generate temporal patterns that characterize the dynamic behavior of the system. Interestingly, these patterns capture the dynamics, which was not the case when producing the diagnosis directly from the classification.

Finally, diagnosis is discussed in relation with theories that participate to provide global solutions to fault management problems. The interactions of diagnosis with other modules, such as planning in autonomous architectures or prognosis in condition maintenance architectures, is key to the efficiency of such architectures. It is indeed the role of diagnosis to estimate the updated health status of the system and its components, henceforth allowing to make relevant and timely decisions. There is still work to be done before achieving perfectly unified frameworks in which diagnosis and the other tasks share the same models or can rely on model transformation technologies to share the same knowledge base. This is an active field of research.

To conclude, diagnosis is concerned with interpreting and explaining the observations acquired on a given system. In AI, diagnosis is often qualified as endowing a system with a "conscience". Diagnosis is not only concerned with faults and faulty situations, but with the state of the system and why it is in that state. This paper points at several opportunities central to the development of the diagnosis field, and that require integrative solutions. These opportunities are numerous and will participate to the richness of the field in the next few years.

References

Ackerson, G., Fu, K., 1970. On state estimation in switching environments. IEEE Transactions on Automatic Control 15, 10–17.

Adrot, O., Maquin, D., Ragot, J., 1999. Fault detection with model parameter structured uncertainties, in: Proceedings of the European Control Conference, ECC'99, Karlsruhe.

Aguilar-Castro, J., Subias, A., L.Travé-Massuyès, Zouaoui, K., 2012. A chronicle learning approach for self-adapting strategies in collaborative communicating systems. LAAS-CNRS Report 11698. LAAS. Toulouse, France, 12p.

Aguilar-Martin, J., Lopez de Mantaras, R., 1999. Approximate reasoning in decision analysis. The Process of Classification and Learning the Meaning of Linguistic Concepts 1982, 165–174.

Alami, R., Chatila, R., Fleury, S., Ghallab, M., Ingrand, F., 1998. An architecture for autonomy. The International Journal of Robotics Research 17, 315–337.

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: Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09, Barcelona, Spain.

Artikis, A., Skarlatidis, A., Portet, F., Paliouras, G., 2010. Logic-based event recognition. Knowledge Engineering Review (to appear) .

Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing 50, 174–188.

Basseville, M., 1988. Detecting changes in signals and systems : a survey. Automatica 24, 309–326.

Basseville, M., Mevel, L., Goursat, M., 2004. Statistical model-based damage detection and localization: subspace-based residuals and damage-to-noise sensitivity ratios. Journal of Sound and Vibration 275, 769–794.

Basseville, M., Nikiforov, I., 1993. Detection of abrupt changes: theory and application. Citeseer.

Bayoudh, M., Travé-Massuyès, L., Olive, X., 2008a. Coupling continuous and discrete event system techniques for hybrid system diagnosability analysis, in: Proceedings of the 18th European Conference on Artificial Intelligence ECAI'08 , July 21-25, 2008, Patras, Greece: Including Prestigious Applications of Intelligent, IOS Press. p. 219.

Bayoudh, M., Travé-Massuyes, L., Olive, X., 2008b. Towards active diagnosis of hybrid systems, in: Proceedings of the 19th workshop on Principles of Diagnosis (DX08), Blue Mountains, Australia, pp. 231–237.

Bayoudh, M., Travé-Massuyès, L., Olive, X., 2009. Active diagnosis of hybrid systems guided by diagnosability properties, in: Proceeding of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09, Barcelona, Spain. pp. 1498–1503.

Bayoudh, M., Travé-Massuyès, L., Olive, X., Space, T., 2008c. Hybrid systems diagnosis by coupling continuous and discrete event techniques, in: Proceedings of the IFAC World Congress, Seoul, Korea, pp. 7265–7270.

Benazera, E., 2009. On the articulation of planning and diagnosis. Technical report 09536, LAAS-CNRS, Toulouse, France.

Benazera, E., Travé-Massuyès, L., 2003. The Consistency approach to the on-line prediction of hybrid system configurations, in: Proceedings of the International Conference on Analysis and design of hybrid systems ADHS'03: a proceedings volume from the IFAC Conference, St. Malo, Brittany, France, 16-18 June 2003, Elsevier Science. p. 241.

Benazera, E., Travé-Massuyès, L., 2009. Set-theoretic estimation of hybrid system configurations. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society 39, 1277–1291.

Benazera, E., Travé-Massuyès, L., Dague, P., 2002. State tracking of uncertain hybrid concurrent systems, in: Proceedings of the 13th International Workshop on Principles of Diagnosis (DX02), pp. 106–114.

Biswas, G., Cordier, M., Lunze, J., Travé-Massuyès, L., Staroswiecki, M., 2004. Diagnosis of complex systems: Bridging the methodologies of the FDI and DX communities. IEEE Transactions on Systems, Man, and Cybernetics, Part B 34, 2159–2162.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2003. Diagnosis and fault-tolerant control. Springer Verlag.

Blom, H., Bar-Shalom, Y., 1988. The interacting multiple model algorithm for systems with markovian switching coefficients. IEEE Transactions on Automatic Control 33, 780–783.

Bregon, A., Daigle, M., Roychoudhury, I., 2012. An integrated framework for model-based distributed diagnosis and prognosis, in: Proceedings of the Annual Conference of the Prognostics and Health Management Society, pp. 416–426.

Chanthery, E., Pencole, Y., Bussac, N., 2010. An ao*-like algorithm implementation for active diagnosis, in: Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2010), Sapporo (Japon). pp. 378–385.

Chow, E., Willsky, A., 1984. Analytical redundancy and the design of robust failure detection systems. IEEE Transactions on automatic control 29, 603–614.

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 Transactions on Systems, Man, and Cybernetics, Part B 34, 2163–2177.

Cram, D., Mathern, B., Mille, A., 2011. A complete chronicle discovery approach: application to activity analysis. Expert Systems , 1–26.

Dague, P., Travé-Massuyès, L., 2004. Raisonnement causal en physique qualitative. Intellectica (38) , 247–290.

Daigle, M., Koutsoukos, X., Biswas, G., 2010. Improving diagnosability of hybrid systems through active diagnosis?, in: Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain.

Daigle, M.J., Goebel, K., 2013. Model-based prognostics with concurrent damage progression processes. Systems, Man, and Cybernetics: Systems, IEEE Transactions on 43, 535–546.

De Jonge, F., Roos, N., Witteveen, C., 2006. Diagnosis of multi-agent plan execution, in: Multiagent System Technologies. Springer, pp. 86–97.

De Kleer, J., Williams, B., 1987. Diagnosing multiple faults. Artificial Intelligence 32, 97–130.

Denoeux, T., Masson, M., Dubuisson, B., 1997. Advanced pattern recognition techniques for system monitoring and diagnosis: a survey. Journal européen des systèmes automatisés 31, 1509–1539.

Dousson, C., Duong, T.V., 1999. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems., in: Proceedings of the 16th Int. Joint Conf. on Artificial Intelligence IJCAI'99, pp. 620–626.

Dousson, C., Gaborit, P., Ghallab, M., 1993. Situation recognition: representation and algorithms, in: Proceedings of the International Joint Conference on Artificial Intelligence IJCAI'93, Chambéry, France. pp. 166–172.

Dubuisson, B., 2001. Automatique et statistiques pour le diagnostic. Hermes Science Europe Ltd .

DX'01, 2001. Bridge workshop. URL: `http://www.di.unito.it/~dx01`.

Feenstra, P., Mosterman, P., Biswas, G., Breedveld, P., 2001. Bond graph modeling procedures for fault detection and isolation of complex flow processes. Simulation Series 33, 77–84.

Fillatre, L., Nikiforov, I., 2007. Non-Bayesian detection and detectability of anomalies from a few noisy tomographic projections. IEEE Transactions on Signal Processing 55, 401.

Finzi, A., Ingrand, F., Muscettola, N., 2004. Model-based executive control through reactive planning for autonomous rovers, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2004, IEEE. pp. 879–884.

Fouladirad, M., Freitag, L., Nikiforov, I., 2008. Optimal fault detection with nuisance parameters and a general covariance matrix. International Journal of Adaptive Control and Signal Processing 22, 431.

Frank, P., 1994. On-line fault detection in uncertain nonlinear systems using diagnostic observers: a survey. International journal of systems science 25, 2129–2154.

de Freitas, N., 2002. Rao-blackwellised particle filtering for fault diagnosis, in: Proceedings of the IEEE Aerospace Conference 2002, pp. 1767–1772.

Fukunaga, K., 1990. Introduction to statistical pattern recognition. Access Online via Elsevier.

Gentil, S., Montmain, J., Combastel, C., 2004. Combining FDI and AI approaches within causal-model-based diagnosis. IEEE Transactions on Systems, Man, and Cybernetics, Part B 34, 2207–2221.

Gertler, J., 1998. Fault Detection and Diagnosis in Engineering Systems. Marcel Deker.

Ghallab, M., Nau, D., Traverso, P., 2004. Automated Planning: theory and practice. Morgan Kaufmann.

Grewal, M., Glover, K., 1976. Identifiability of linear and nonlinear dynamical systems. IEEE Transactions on Automatic Control 21, 833–837.

Henzinger, T., 1996. The theory of hybrid automata, in: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96), New Brunswick, New Jersey. pp. 278–292.

Hofbaur, M., Williams, B., 2002a. Hybrid diagnosis with unknown behavioral modes, in: Proceedings of the Thirteenth Internatinal Workshop on Principles of Diagnosis (DX-02), Semmering, Austria, pp. 97–105.

Hofbaur, M.W., Williams, B.C., 2002b. Mode estimation of probabilistic hybrid systems, in: Tomlin, C., Greenstreet, M. (Eds.), Hybrid Systems: Computation and Control, HSCC 2002. Springer Verlag. volume 2289 of *Lecture Notes in Computer Science*, pp. 253–266.

Hofbaur, M.W., Williams, B.C., 2004. Hybrid estimation of complex systems. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 34, 2178–2191.

Jauberthie, C., Chanthery, E., 2013. Optimal input design for a nonlinear dynamical uncertain aerospace system, in: Proceedings of the 9th IFAC Symposium on Nonlinear Control Systems, Toulouse, France.

Jauberthie, C., Verdière, N., Travé-Massuyès, L., 2011. Set-membership identifiability: Definitions and analysis, in: Proceedings of the IFAC World Congress, Milan, Italy. pp. 12024–12029.

Katsillis, G., Chantler, M., 1997. Can dependency-based diagnosis cope with simultaneous equations, in: Proceedings of the 8th International Workshop on Principles of Diagnosis DX-97, pp. 51–59.

Kempowsky, T., Aguilar-Martin, J., A.Subias, Lann, M.L., 2003. Classification tool based on interactivity between expertise and self-learning

techniques, in: Proceedings of the International Symposium on Fault Detection, Supervision and Safety of Technical Processes Safeprocess 2003, Washington D.C. (USA).

Kempowsky, T., Subias, A., Aguilar-Martin, J., 2006a. Process situation assessment: From a fuzzy partition to a finite state machine. Engineering Applications of Artificial Intelligence 19, 461–477.

Kempowsky, T., Subias, A., Aguilar-Martin, J., Travé-Massuyès, L., 2006b. A discrete event model for situation awareness purposes, in: Proceedings of the International Symposium on Fault Detection, Supervision and Safety of Technical Processes Safeprocess 2006, Beijing, China. pp. 1288–1293.

Kieffer, M., Walter, E., 2011. Guaranteed estimation of the parameters of nonlinear continuous-time models: Contributions of interval analysis. International Journal of Adaptive Control and Signal Processing 25, 191–207.

Kleer, J., Mackworth, A., Reiter, R., 1992. Characterizing diagnoses and systems. Artificial Intelligence 56, 197–222.

Kramer, M., Palowitch, B., 1987. A rule-based approach to fault diagnosis using the signed directed graph. AIChE Journal 33, 1067–1078.

Krysander, M., Aslund, J., Nyberg, M., 2008. An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 38, 197–206.

Kuhn, L., Price, B., De Kleer, J., Do, M.B., Zhou, R., 2008. Pervasive diagnosis: The integration of diagnostic goals into production plans., in: Proceedings of the AAAI, pp. 1306–1312.

Kurien, J., Nayak, P.P., 2000. Back to the future for consistency-based trajectory tracking, in: Proceedings of the AAAI/IAAI Conference, pp. 370–377.

Lerner, U., Moses, B., Scott, M., McIlraith, S., Koller, D., 2002. Monitoring a complex physical system using a hybrid dynamic bayes net, in: Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence UAI'02.

Lerner, U., Parr, R., Koller, D., Biswas, G., 2000. Bayesian fault detection and diagnosis in dynamic systems, in: Proceedings of the AAAI/IAAI Conference, pp. 531–537.

Leyval, L., Gentil, S., Feray-Beaumont, S., 1994. Model-based causal reasoning for process supervision. Automatica 30, 1295–1306.

L.Hedjazi, Lann, M., T.Kempowsky, J.Aguilar-Martin, F.Dalenc, G., L. Despenes, S., 2011. From chemical process diagnosis to cancer prognosis: an integrated approach for diagnosis and sensor/marker selection, in: Proceedings of the European Symposium on Computer-Aided Process Engineering (ESCAPE 21), Chalkidiki (Grce), pp. 1510–1514.

Li, X., Bar-Shalom, Y., 1996. Multiple-model estimation with variable structure. IEEE Transactions on Automatic Control 41, 478–493.

Liu, F., Qiu, D., 2008. Safe diagnosability of stochastic discrete event systems. Automatic Control, IEEE Transactions on 53, 1291–1296.

Loiez, E., Taillibert, P., 1997. Polynomial temporal band sequences for analog diagnosis, in: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence IJCAI-97, Nagoya, Japan, August 23-29, 1997, p. 474.

Lunze, J., (Eds.), F.L.L., 2009. Handbook of hybrid systems control: theory, tools, applications. Cambridge.

Maiga, M., Ramdani, N., Travé-Massuyès, L., 2013. A fast method for solving guard set intersection in nonlinear hybrid reachability, in: Proceedings of the 52nd IEEE Conference on Decision and Control, Florence, Italy (to appear).

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

Mehra, R., 1974. Optimal input signals for parameter estimation in dynamic systems–survey and new results. Automatic Control, IEEE Transactions on 19, 753–768.

Micalizio, R., Scala, E., Torasso, P., 2011. Intelligent supervision for robust plan execution, in: AI* IA 2011: Artificial Intelligence Around Man and Beyond. Springer, pp. 151–163.

Micalizio, R., Torasso, P., 2007. Plan diagnosis and agent diagnosis in multi-agent systems, in: AI* IA 2007: Artificial Intelligence and Human-Oriented Computing. Springer, pp. 434–446.

Micalizio, R., Torasso, P., 2009. Agent cooperation for monitoring and diagnosing a map, in: Multiagent System Technologies. Springer, pp. 66–78.

Milanese, M., Vicino, A., 1991. Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. Automatica 27, 997–1009.

Mohanty, S., Chattopadhyay, A., Peralta, P., 2010. Adaptive residual useful life estimation of a structural hotspot. Journal of Intelligent Material Systems and Structures 21, 321–335.

MONET, . Network of excellence monet II. URL: `http://monet.aber.ac.uk:8080/monet/index.html`.

Mosterman, P., Biswas, G., 1999. Diagnosis of continuous valued systems in transient operating regions. IEEE Transactions on Systems, Man, and Cybernetics, Part A 29, 554–565.

Narasimhan, S., Biswas, G., 2002. An approach to model-based diagnosis of hybrid systems, in: Tomlin, C., Greenstreet, M. (Eds.), Hybrid Systems: Computation and Control, HSCC 2002. Springer Verlag. volume 2289 of *Lecture Notes in Computer Science*, pp. 308–322.

Narasimhan, S., Dearden, R., Benazera, E., 2004a. Combining particle filters and consistency based approaches for monitoring and diagnosis of stochastic hybrid systems, in: Proceedings of the 15th International Workshop on Principles of Diagnosis (DX04), pp. 123–128.

Narasimhan, S., Dearden, R., Benazera, E., 2004b. Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems, in: Proceedings of the15th International Workshop on Principles of Diagnosis (DX04), Carcassonne, France, Citeseer.

Nayak, P., Kurien, J., 2000. Back to the future for consistency-based trajectory tracking, in: Proceedings of the 17th National Conference on Artificial Intelligence AAAI'00, Austin, Texas.

Omlin, C.W., Thornber, K.K., Giles, C.L., 1998. Fuzzy finite-state automata can be deterministically encoded into recurrent neural networks. Fuzzy Systems, IEEE Transactions on 6, 76–89.

Orchard, M.E., Vachtsevanos, G.J., 2009. A particle-filtering approach for on-line fault diagnosis and failure prognosis. Transactions of the Institute of Measurement and Control 31, 221–246.

Oyeleye, O.O., Kramer, M.A., 1988. Qualitative simulation of chemical process systems: Steady-state analysis. AIChE Journal 34, 1441–1454.

Paoli, A., Lafortune, S., 2005. Safe diagnosability for fault-tolerant supervision of discrete-event systems. Automatica 41, 1335–1347.

Patton, R., Chen, J., 1991. A re-examination of the relationship between parity space and observer-based approaches in fault diagnosis. European Journal of Diagnosis and Safety in Automation 1, 183–200.

Patton, R., Frank, P., Clark, R., 1989. Fault diagnosis in dynamic systems. Theory and Applications .

Pell, B., Bernard, D., Chien, S., Gat, E., Muscettola, N., Nayak, P., Wagner, M., Williams, B., 1998. An autonomous spacecraft agent prototype. Autonomous Robots 5, 29–52.

Pinedo, M., 2012. Scheduling: theory, algorithms, and systems. Springer.

Pouliezos, A., Stavrakakis, G., Lefas, C., 1985. Fault detection using parameter estimation. Quality and reliability engineering international 5, 283–290.

Pulido, B., Gonzalez, C., 2004a. Possible conflicts: a compilation technique for consistency-based diagnosis. IEEE Transactions on Systems, Man, and Cybernetics, Part B 34, 2192–2206.

Pulido, B., Gonzalez, C., 2004b. Possible conflicts: A compilation technique for consistency-based diagnosis. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 34, 2192–2206.

Qiu, Z., Gertler, J., 1993. Robust FDI and H_{inf} optimization, in: Proceedings of the 32nd IEEE Conference on Control and Decision CDC'93, San Antonio, Texas.

Ramadge, P.J., Wonham, W.M., 1989. The control of discrete event systems. Proceedings of the IEEE 77, 81–98.

Ramdani, N., Nedialkov, N.S., 2011. Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques. Nonlinear Analysis: Hybrid Systems 5, 149–162.

Reiter, R., 1987. A theory of diagnosis from first principles. Artificial Intelligence 32, 57–95.

Ribot, P., Pencolé, Y., Combacau, M., 2008. Prognostics for the maintenance of distributed systems, in: Proceedings of the International Conference on Prognostics and Health Management (PHM'08), Denver, USA, pp. 6–10.

Ribot, P., Pencolé, Y., Combacau, M., 2009. Diagnosis and prognosis for the maintenance of complex systems, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics SMC 2009, IEEE. pp. 4146–4151.

Rienmüller, T., Hofbaur, M., Travé-Massuyès, L., Bayoudh, M., 2013. Mode set focused hybrid estimation. International Journal of Applied Mathematics and Computer Science 23, 131–144.

Rienmuller, T., Mayoudh, M., Hofbaur, M., Travé-Massuyès, L., 2009. Hybrid estimation through synergic mode-set focusing, in: Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09.

Russell, S.J., Norvig, P., Canny, J.F., Malik, J.M., Edwards, D.D., 1995. Artificial intelligence: a modern approach. volume 74. Prentice hall Englewood Cliffs.

Sachenbacher, M., Williams, B., 2004. Diagnosis as semiring-based constraint optimization, in: Proceedings of the European Conference on Artificial Intelligence ECAI'04, p. 873.

Sampath, M., Lafortune, S., Teneketzis, D., 1998. Active diagnosis of discrete-event systems. Automatic Control, IEEE Transactions on 43, 908–929.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D., 1995. Diagnosability of discrete-event systems. IEEE Transactions on Automatic Control 40, 1555–1575.

Saxena, A., Celaya, J., Saha, B., Saha, S., Goebel, K., 2010. Metrics for offline evaluation of prognostic performance. International Journal of Prognostics and Health Management 1, 20.

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

Staroswiecki, M., Declerck, P., 1989. Analytical redundancy in non linear interconnected systems by means of structural analysis., in: Proceedings of the IFAC Symposium on Advanced Information Processing in Automatic Control, pp. 51–55.

Subias, A., Exposito, E., Chassot, C., Travé-Massuyès, L., Drira, K., 2010. Self-adapting strategies guided by diagnosis and situation assessment in collaborative communicating systems, in: Proceedings of the International Workshop on Principles of Diagnosis (DX 10), Portland (USA), pp. 329–336.

Thrun, S., Langford, J., Verma, V., 2002. Risk sensitive particle filters, in: Proceedings of the Conference on Advances in neural information processing systems, MIT Press. p. 961.

Travé-Massuyès, L., Calderon-Espinoza, G., 2007. Timed fault diagnosis, in: Proceedings of the European Control Conference ECC-07, Kos, Greece.

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

Travé-Massuyès, L., Pons, R., Tornil, S., Escobet, T., 2001. The CA-En Diagnosis System and its Automatic Modelling Method. Computación y Sistemas 5, 128–143.

AFIA, . Association française d'intelligence artificielle. URL: `http://www.afia.asso.fr/`.

BRIDGE, . Bridging AI and control engineering model-based diagnosis approaches. URL: `http://monet.aber.ac.uk:8080/monet/monetinfo/monetbridge.htm`.

GDR-I3, . Groupement de recherche information, interaction, intelligence. URL: `http://www.irit.fr/GDR-I3/`.

GDR MACS, . Groupement de recherche modélisation, analyse et conduite des systèmes dynamiques. URL: `http://www.univ-valenciennes.fr/GDR-MACS/`.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., Wu, B., 2006. Intelligent fault diagnosis and prognosis for engineering systems. Wiley Online Library.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. Computers & Chemical Engineering 27, 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003b. A review of process fault detection and diagnosis: Part iii: Process history based methods. Computers & Chemical Engineering 27, 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003c. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. Computers & chemical engineering 27, 293–311.

Vento, J., Puig, V., Sarrate, R., Travé-Massuyès, L., 2010. Fault detection and isolation of hybrid systems using diagnosers that reason on components, in: Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (Safeprocess'2012), Mexico City, Mexico, IFAC. pp. 1250–1255.

Verfaillie, G., Charmeau, M.C., 2006. A generic modular architecture for the control of an autonomous spacecraft, in: Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS), Space Telescope Science Institute Baltimore, MD, USA.

Verma, V., Gordon, G., Simmons, R., Thrun, S., 2004. Real-time fault diagnosis. IEEE Robotics and Automation Magazine 11, 56–66.

Verma, V., Thrun, S., Simmons, R., 2003. Variable resolution particle filter, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'03), pp. 976–981.

Vilain, M.B., Kautz, H.A., 1986. Constraint propagation algorithms for temporal reasoning., in: Proceedings of the 5th National Conference on Artificial Intelligence AAAI'86, pp. 377–382.

Washington, R., 2000. On-board real-time state and fault identification for rovers, in: Proceedings of the IEEE International Conference on Robotic and Automation (ICRA'00), San Francisco, CA, USA.

Washio, T., Motoda, H., Niwa, Y., INSS, I., 1999. Discovering Admissible Model Equations from Observed Data, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence IJCAI'99, Citeseer. pp. 772–779.

Weld, D., De Kleer, J., 1989. Readings in qualitative reasoning about physical systems. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Williams, B., Nayak, P., 1996. A model-based approach to reactive self-configuring systems, in: Proceedings of the National Conference on Artificial Intelligence, pp. 971–978.

Williams, B., Ragno, R., 2003. Conflict-directed A* and its role in model-based embedded systems, in: Journal of Discrete Applied Mathematics, Citeseer.

Williams, B.C., Ingham, M.D., Chung, S.H., Elliott, P.H., 2003. Model-based programming of intelligent embedded systems and robotic space explorers. Proceedings of the IEEE 91, 212–237.

Witteveen, C., Roos, N., van der Krogt, R., de Weerdt, M., 2005. Diagnosis of single and multi-agent plans, in: Proceedings of the 4th International Joint Conference on Autonomous agents and multiagent systems, ACM. pp. 805–812.

Zaytoon, J., et al., 2001. Systèmes dynamiques hybrides. Hermès Science Publication, Paris .