PRACTICAL CONSIDERATIONS IN SYSTEMS DIAGNOSIS USING TIMED FAILURE PROPAGATION GRAPH MODELS

Sherif Abdelwahed **Gabor Karsai**

Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN 37203

Abstract - Timed failure propagation graphs (TFPG) are causal models that capture the temporal aspects of failure propagation in dynamic systems. In this paper we present several practical modeling and reasoning considerations that have been addressed based on experience with avionics systems. These include the problem of intermittent faults, handling test alarms, dealing with limited computational resources, and model reductions for large scale systems.

INTRODUCTION

The ability to react rapidly to unpredictable and dramatic changes in both the system and the environment is a key requirement for future engineering systems and applications. Successful operation of these systems relies on the existence of efficient mechanisms to detect, isolate and correct problems early, and to re-allocate resources where they are most needed. Model-based diagnoses technologies help address these issues by enabling run-time analysis and providing effective means to identify operation problems in the early stages.

In earlier work [1] we developed a qualitative condition-based diagnosis approach based on the timed failure propagation graph (TFPG) model aimed to represent the failure behavior of a general class of practical sensor-based engineering systems. Timed failure propagation graphs are causal models that describe the system behavior in presence of faults. The TFPG structure captures the effect of timing constraints and switching dynamics on the propagation of failures in practical discrete event and hybrid systems. The TFPG model is closely related to the fault model presented in [8][10] and used in an integrated fault diagnosis and process control system. The temporal aspects of the TFPG model are closely related to the domain theoretic notion of temporal dependency proposed in [4]. However, TFPG based diagnosis is an online incremental approach that focuses on diagnosis robustness with respect to failures in the alarms (i.e., missing or spurious). In addition, we consider a specific form of temporal dependency that directly incorporates the effect multi-mode dynamics on failure propagation.

In [2] we presented a consistency-based approach for robust diagnosis of systems in which failure behavior can be captured by TFPG models. This proposed diagnosis algorithm is executed online and is based on incremental non-monotonic reasoning that is robust with respect to various forms of alarm failures. The proposed algorithm consists of two main procedures; the first one generates an optimal consistent initial state assignment based on current state observation while the second procedure generates the set of all consistent hypotheses from a given initial state assignment.

In this paper we present several practical modeling and reasoning considerations that have been addressed based on an experience with avionics systems [9]. In particular, we address the problem of intermittent faults, handling test alarms, dealing with limited computational resources, and model reductions for large scale systems. We discuss the practical context of these problems and introduce the model and reasoning extensions that are developed to handle these issues. Although, these issues have been tackled in the context of an avionics system, the proposed approaches are applicable to a wide variety of practical engineering systems.

Intermittent faults have been addressed in literature for various diagnosis models. In [7], an extension to the conventional model-based diagnosis algorithms is introduced to handle intermittent faults. The work in [11] converts intermittent diagnosis tasks of combinational logic to dynamic programming. In [5] an intermittent diagnosis approach is presented for a general class of discrete event systems which assumes intermittent faults are followed by the corresponding reset event. This assumption is relaxed in [13] for a single intermittent faults or spurious alarm. Diagnosis of intermittent failure for industrial process modeled as Petri-nets are introduced in [6].

The paper is organized as follows. In Section 2, the timed failure propagation graph model is introduced. Section 3 presents an overview of the consistency based diagnosis approach for TFPG. Section 4 introduces the issues of intermittent faults, test alarms, reasoning complexity, and model reductions. This section also introduces the extensions to the TFPG modeling structure and reasoning algorithms proposed to handle these issues. Conclusion and future works are presented in Section 5.

TIMED FAILURE PROPAGATION GRAPHS

A TFPG is a labeled directed graph where the nodes represent either failure modes, which are the causes, or discrepancies, which are off-nominal conditions that are the effects of failure modes. Edges between nodes in the graph capture propagation of failure effects over time in the dynamic system. To represent of failure propagation in multi-mode (switching) systems, edges in the graph model can be activated or deactivated depending on a set of possible operation modes of the system. Formally, a TFPG is represented as a tuple G = (F, D, E, M, ET, EM, DC, DS), where:

- F is a nonempty set of failure nodes.
- D is a nonempty set of discrepancy nodes.
- $E \subseteq V \times V$ is a set of edges connecting the set of all nodes $V = F \cup D$. src(e) and dst(e) denotes the source and destination nodes of the edge e, respectively.
- M is a nonempty set of system modes. At each time instance t the system can be in only one mode.
- ET: $E \rightarrow (\Re \times \Re)$ is a map that associates every edge in E with a time interval.
- EM: $E \rightarrow \wp(M)$ is a map that associates every edge in E with a set of modes in M. We assume that $EM(e) \neq \emptyset$ for any edge $e \in E$.
- DC: $D \rightarrow$ {AND, OR} is a map defining the class of each discrepancy as either AND or an OR node.
- DS: $D \rightarrow$ {M, U} is a map defining the monitoring status of the discrepancy as either M for the case when the discrepancy is monitored by an online alarm or U for the case when the discrepancy is not monitored.

The set V contains n + m vertices, representing n failure modes and m discrepancies. The map ET associates each edge $e \in E$ with the minimum and maximum time for the failure to propagate along the edge. For an edge $e \in E$, we will use the notation e.tmin and e.tmax to indicate the corresponding minimum and maximum time for failure propagation along e, respectively. That is, given that a propagation edge is enabled (active), it will take at least (most) tmin (tmax) time for the fault to propagate from the source node to the destination node. The map EM associates each edge $e \in E$ with a subset of the system modes at which the failure can propagate along the edge. Consequently, the propagation link e is enabled (active) in a mode $m \in M$ if and only if $m \in EM(e)$. The map DC defines the type of a given discrepancy as either AND or OR. An OR type discrepancy node will be activated when the failure propagate to the node form any of its parents. On the other hand, an AND discrepancy node can only be activated if the failure propagates to the node from all its parents. We assume that TFPG models do not contain self loops and that failure modes are always root nodes, i.e., they cannot be a destination of any edge. Also, a discrepancy cannot be a root node, that is, every discrepancy must be a successor of another discrepancy or a failure mode.

Figure 1 shows an example of failure propagation graph model. Rectangles in the graph model represent the failure modes while circles and squares represent OR and AND type discrepancies, respectively. The arrows between the nodes represent failure propagation. Propagation edges are parameterized with the corresponding interval, [e.tmin, e.tmax], and the set of modes at which the edge is active. The above figure shows also a sequence of alarm signals identified by shaded discrepancies. The time at which the alarm is observed is shown above the corresponding discrepancy.

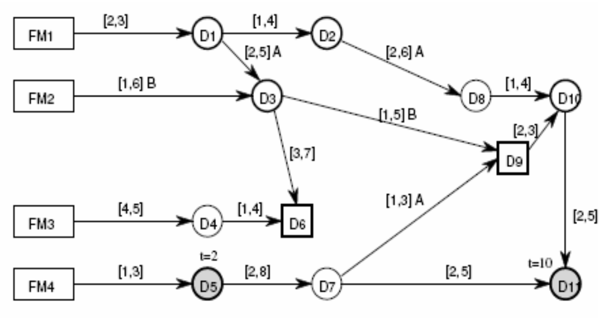

Figure 1: A timed failure propagation graph

The TFPG model captures observable failure propagations between discrepancies in practical systems. In this setting, alarms capture state deviations from nominal values. The set of all observed deviations corresponds to the discrepancy set in the TFPG model. Propagation edges, on the other hand, correspond to causality (as defined by energy flow, for instance) in the system dynamics. Due to the dynamic nature of the system, failure effects take time to propagate between the system's components. Such time in general depends on the system's time constants as well as the size and timing of underlying failure. Propagation delay intervals can be computed analytically or through simulations using an accurate model.

Failure propagation in a TFPG has a simple semantics. The state of a node indicates if the failure effects reached this node. Failure effects can reach a node from any of its predecessors, if it is an OR type node. Failure can only reach nodes of AND type only if it already reached all its parents. For an OR type node v' and an edge $e = (v,v') \in E$, once a failure effect reaches v at time t it must reaches v' at a time t' where $e.tmin \leq t' - t \leq e.tmax$. On the other hand, the activation period of an AND alarm v' is the min/max composition of the activation periods for each link $(v,v') \in E$. For a failure to propagate through a link (v,v'), the link should be active throughout the propagation, that is, from the time the failure reaches v to the time it reaches v'. If the link is deactivated any time during the propagation (because of mode switching), the propagation stops. Links are assumed to be memoryless with respect to failure propagation so that current failure propagation is independent of any (incomplete) previous propagation. Also, once a failure effect reaches a node its state will change permanently, and it will not be affected by any future failure propagation.

REVIEW OF THE DIAGNOSIS APROACH

An *actual (physical) system* state corresponds to the current state of all nodes in the TFPG model. A physical state a time t is given by a map AS_t: $V \rightarrow \{ON, OFF\} \times \Re$, V is the set of nodes in the TFPG model. An ON state for a node indicates that the failure (effect) reached this node, otherwise it is set to OFF. The state at time t is denoted $AS_t(v)$.state, while $AS_t(v)$.time denote the last time at which the state of v is changed. Failure effects are assumed permanent; therefore, the state of a node once changed will remain constant after that. Due to mode switching, a similar map is used to define the state of edges, namely, ES_t: $E \rightarrow \{ON, OFF\} \times \Re$ with $ES_t(e)$.state defines the state of the edge e at time t, while $ES_t(e)$.time is the last time at which the state of e is changed.

The observed state of the system may not be consistent with the failure propagation graph model temporal constraints, due to potential alarm failures. The *observed state* at time t is defined as a map S_t: $D \rightarrow \{ON, OFF\} \times \Re$. Clearly, observed states are only defined for discrepancies. We assume that alarm signals are permanent so that the observed state of a discrepancy once changed will remain constant after that. This assumption of permanent change also applies to faulty alarms.

The aim of the diagnosis reasoning process is to find a consistent and plausible explanation of the current system state based on the observed state. Such explanation is given in form of a valid hypothetical state. A *hypothetical state* is a map that defines node states and the interval at which each node changes its state. Formally a hypothetical state at time t is a map H_t: $V \rightarrow \{ON, OFF\} \times \Re \times \Re$. Similar to actual states, hypothetical states are defined for both discrepancies and failure modes. The estimated earliest (latest) time of state change is denoted H(v).terl (H(v).tlat).

A hypothetical state is an estimation of the current state of all nodes in the system and the point in time at which this node changed its states. An estimation of the current state is valid only if it is consistent with the TFPG model. State consistency in TFPG models is a node-parents relationship that can be extended pairwise to arbitrary subsets of nodes. State consistency can be defined based on the semantics of failure propagation, and it can be used to check if an OR (AND) alarm is consistent with any of (all) its parent. Because of mode switching, the node-parents consistency depends not only on the state of the underlying nodes but also on the current state of the connection edges and their last activation times. Formally, let $d \in D$ be an OR type discrepancy. Then, a hypothesis map H_t is *consistent* with respect to d if:

1. $H_t(d)$ = OFF and for all $(v,d) \in E$:
   a. $H_t(v)$ = OFF, or
   b. $H_t(v)$ = ON and $ES_t(v,d)$.state = ON and $t < \max(H_t(v).tlat, ES_t(e).time) + (v,d).tmax$
2. $H_t(d)$ = ON and all the following hold:
   a. $H_t(d).terl \geq \min_{v \in Ud} \{ H_t(v).terl + (v,d).tmin \}$,
   b. $H_t(d).tlat \leq \min_{v \in Ud} \{ H_t(v).tlat + (v,d).tmax \}$

Where $U_d = \{ v \in V \mid (v,d) \in E$ and $H_t(v).state$ = ON

The consistency relationship is the foundation of the diagnosis approach presented in [2]. This relationship is used to check the consistency of observed discrepancies. It is also used to generate a maximal (with respect to activation period) hypothetical state for an alarm given the hypothetical state of its parents. Ultimately, this can be used to extend a hypothetical state defined for a subset $V' \subseteq V$ through forward-propagation up to the leaf nodes. Extension through backward-propagation of hypothetical states can also be defined based on the consistency relationship.

The optimal diagnosis algorithm (optimality here is defined with respect to the size of the ambiguity set) consists of the following main steps:

- Compute the maximal set(s) of consistent discrepancy with respect to the current observed state.
- Generate a consistent partial hypothetical state covering the observed discrepancy.
- Propagate the generated partial hypothetical states backward up to the parent set of failure mode. All other failure modes are assigned the default OFF state.
- Propagate the new partial hypothetical state forward recursively until all nodes in the TFPG model are covered.

This procedure is conducted incrementally based on previous computation at the occurrence of every state-changing event (alarm signal or time-out). A failure report is then generated from the computed set of optimal hypotheses. The failure report enlists the set of all consistent state assignments that maximally matches the current set of observation. Any observed state that does not match the current hypothesis is considered faulty. A detailed description and analysis of the diagnosis algorithm can be found.

Hypotheses Ranking

The quality of hypotheses in the TFPG settings can be measured based on two independent factors Hypotheses are evaluated based on the following metrics:

- Plausibility: reflects the support of a hypothesis based on the current observed alarm state. It typically answers the user question: Which hypothesis to consider?
- Robustness: reflects the potential of a hypothesis (evidence) to change based on remaining alarms (that are downstream in the path and could be activated in the future). It typically answers the user question: When to take an action?

The plausibility metric considers two independent factors, namely, alarm consistency and failure mode parsimony. The alarm consistency factor is defined as the ratio of the active consistent alarms to that of all (currently) identified alarms. The failure mode factor is defined as the ratio of identified failure modes (according to the underlying hypothesis) to the total number of failure modes in the system. This factor is a direct representation of the parsimony principle (less failures is more plausible). Hypotheses plausibility are ordered lexicographically (alarm factor is more dominant).

The diagnoser updates the current set of hypothesis incrementally in a way to improve the current plausibility measure. In other words, the diagnoser will consider changes to the hypothetical states only if such changes can increase the plausibility of the underlying hypothesis. In addition changes are restricted so that the changed hypothesis remains valid.

PRACTICAL CONSIDERATIONS IN TFPG BASED DIAGNOSIS

The original TFPG diagnosis tool was developed for a general class of engineering applications. Since then major updates has been made to address the complexity of applications dynamics and the limitations of practical execution platforms. In this section we introduce several practical modeling and reasoning issues that have been addressed based on our experience with avionics system applications.

Intermittent faults

Intermittent faults are common in practical engineering systems. They correspond to situations in which some of the failure conditions are observed for a bounded time interval. Intermittent faults can be caused by sensors inaccuracy, temporary changes in the system components or its operation conditions, external disturbances, and signal noise. If not considered in the diagnosis systems, intermittent faults can lead to decrease in the diagnosis accuracy (ambiguity set increase, reducing the ranking difference) or even false diagnosis.

The original TFPG diagnosis engine assumes permanent state change for monitored discrepancies. That is, once an alarm is activated it will remain activated throughout the operation cycle of the system. Assuming all faults to be permanent, the steps leading from fault identification to fault isolation to fault recovery though a reconfiguration of the architecture are rather consolidated in practice.

The situation can be more complicated in the presence of intermittent faults that need to be distinguished and handled accordingly. In some practical situations, intermittent faults are treated as permanent faults until confirmed otherwise. This is possible in situations where the fault recovery procedure is simple and the system is capable of reconfiguring itself while in operation. In other situations, such easy recovery is not possible and the only alternative is to wait until the nature of the fault is confirmed.

In sensor based systems, an intermittent fault will cause a set of alarms to turn OFF after a period of activation. As mentioned earlier this can lead to significant decrease in the accuracy of the diagnosis. Such decrease depends on the average period and frequency and average duration of intermittent faults. It can be shown that high frequency intermittent faults can lead to information loss which can lead to irreversible (online) diagnosis inaccuracy. However, for the typical situation of low frequency intermittent faults, it is possible to recover the accurate diagnosis. This recovery can be done after all intermittent faults are identified. In this case, diagnosis errors can be corrected by backtracking all the diagnosis results up to the last accurate point and re-computing diagnosis forward from this point up to the current instance.

Intermittent faults can be handled directly at run time by backtracking when the monitored alarm changes its state from ON to OFF. In more concrete terms, let d_i be an intermittent alarm that change its state from ON at time t_1 to OFF at time t_2, let Alarms(t_1, t_2) be the list of events the form e= (d, t) where d is an alarm that is activated (to an ON state) at time t where $t_1 \leq t \leq t_2$. We will write H' = Diag(H_t,e) for the set of hypotheses generated at time t in response to an event e= (d, t) starting with a set of hypotheses H_t (generated in response to previous events). We can extend the function Diag to a list of events in the usual way. According to the above, the diagnoser will handle the intermittent fault d by computing the following set of hypotheses at time t_2,

$$H' = \mathrm{Diag}(H_{t1}, \mathrm{Alarms}(t_1, t_2) - \{(d,t_1)\})$$

Clearly, in order to perform the above computations the diagnosis system needs to keep track of the set H_t for all time t in which an event e = (d,t) occurred, as well as the set Alarms(0,t) of all alarms triggered up to the current time t.

The complexity of the above procedure depends directly on the size of the set Alarms(t_1, t_2) as well as the size of H_{t1} which typically grows monotonically with t_1. Therefore there could be a significant overhead associated with long-duration intermittent fault or those who appear later in the execution cycle of the system. To reduce the complexity of this procedure, the diagnoser identifies potential intermittent alarms (for instance, those with relatively little support from other alarms) and considers branching (two different diagnosis paths) for both states (ON/OFF). This approach can reduce peak processing time. However, it will not affect the average processing time, and in general it will require more memory. The identification of intermittent alarms can be either through a user defined criterion (failure rate) or by learning from previous history of the system. Another way to reduce the complexity of dealing with intermittent alarms is to delay the diagnosis procedure for a limited time until potential intermittent alarms stabilize. This can be applied in situations where intermittent duration is relatively short and the system can tolerate the transient state.

Test Alarms

Test alarms are a set of sensors that can be enabled and disabled by the user, and when they are disabled their status is unknown. In contrast, regular sensors are always enabled during the operation cycle of the system and their state is always known (ON or OFF). Test alarms are used for offline maintenance but can also be used online to check certain failure conditions (hypotheses) as reported by the system diagnoser. A test alarm can be enabled and disabled several times during the normal operation of the system. The value of a test alarm value can only be used in diagnosis if it is enabled. That is a disabled test alarm will not be considered when computing any of the diagnosis metrics (plausibility and robustness).

To handle test alarms in the TFPG diagnosis system, the map DS is extended with the test alarm type. Accordingly, DS is defined as DS: D $\rightarrow$ {M, U, T} where T indicates a test alarm. In addition, the concept of physical state is extended to take into account the enabling condition of the test alarm. When a test alarm is disabled it is treated as unmonitored alarm. An unmonitored alarm is assigned a hypothetical state that is (maximally) consistent the hypothetical states of its parents. Note that the hypothetical state of a monitored alarm is either equal to its physical state or defined in such a way to be consistent with the hypothetical state of its parents (see [2] for a detailed discussion of the consistency-based diagnosis approach).

An enabled test alarm is more complicated, as it cannot be treated as a monitored alarm. This is due to the fact that test alarms do not possess the time information associated with monitored alarms. In particular, if the test alarm is enabled at time t and respond with an ON state, we cannot be sure about the time at which the failure propagates to the associated discrepancy. For this we consider test alarm as time ambiguous. Following the parsimony principle, the following conditions are used to reasoning about test alarm.

- The test alarm d is enabled at time t and respond with ON state:
    - If the ON state of d is consistent with respect to the hypothetical state of its parents for a time t' such that $t_o < t' \leq t$ where t_o is the last activation time of the test alarm, then d is assigned the consistent hypothetical state.
    - Else, d is assumed an inconsistent alarm and is assigned the maximal consistent hypothetical state.
- The test alarm d is enabled at time t and respond with OFF state:
    - If the OFF state of d is inconsistent with respect to the hypothetical state of its parents, then the alarm is assigned a maximally consistent hypothetical state with respect to its parents' hypothetical states. Given the discrepancy between the physical and hypothetical state of d, d will be treated as an inconsistent alarm.
    - Else, d is assumed consistent with respect to its parents.

The above procedure for handling test alarms does not have a major effect on the complexity of reasoning. In practice, however, the ambiguity associated with test alarms may allow some false alarms to pass which can lead to larger number of hypotheses (explanations) than the normal case.

Due to the ambiguity associated with test alarms, it may be desirable to distinguish between test alarms and normal once when considering their contribution to the plausibility of the underlying hypothesis. In particular, test alarms can be assigned a lower weight (defined by the user) in the plausibility formula.

Resource limited reasoning

Certain aspects of the diagnoses systems can be tuned to reduce the run-time computational requirements. These include the max number of hypotheses and the maximum duration of backtracking. Using simulation, it is possible to have approximate estimation of the effect of changing these values on the accuracy and the computations requirements (time and memory) of the diagnosis system such that the diagnoser can be adjusted for specific time and space requirements.

The TFPG diagnosis system provides several approaches to handle platforms with limited resources. These include pre-compilation of reachability and connectivity information that are frequently used throughout the diagnosis process. In additions several heuristics are implemented to limit the number of generated hypotheses by focusing on the most likely ones throughout the reasoning procedures.

Model Reduction

In some situations it is possible to generate a reduced model that is semantically equivalent to a given model. Two main approaches for model can be applied to TFPG models. The first approach is based on the elimination of silent discrepancies. This reduction method will generate a TFPG model with all monitored discrepancies that is semantically equivalent to a given TFPG. The other approach will consider an equivalence relation on the set of discrepancies based on their contribution to the failure identification procedure. Under such equivalence, it is possible to replace the model with a semantically equivalent reduced model with a well defined map between the discrepancies in the original model and those in the reduced one. The conditions for the existence of such equivalent relation can be defined based on the diagnosability metrics introduced in [3]. In many practical situations, significant decrease in both time and space requirements can be achieved through model reduction.

CONCLUSION

In this paper we presented several practical considerations for diagnosing sensor-based systems using timed failure propagation graphs. We discussed the problem of intermittent failures and how the TFPG modeling and reasoning approach is extended using backtracking to handle this class of failure under the assumption of an observable reset condition to follow the underlying spurious alarm. Methods to reduce the complexity of backtracking are also discussed. We introduced also the extensions used to handle test alarms. Finally, we briefly discussed the problems on limited resource and model reductions.

REFERENCES

[1] S. Abdelwahed, G. Karsai and G. Biswas, “System Diagnosis using Hybrid Failure Propagation Graphs”, Proceedings of the Workshop on Principles of Diagnosis (DX-04), Carcassonne, France, 2004.

[2] S. Abdelwahed, G. Karsai, and G. Biswas, “A consistency-based robust diagnosis approach for temporal causal systems,” in The 16th International Workshop on Principles of Diagnosis, Pacific Grove, CA, 2005.

[3] S. Abdelwahed, G. Karsai, and G. Biswas, “Notions of Diagnosability for Timed Failure Propagation Graphs,” in IEEE Systems Readiness Technology Conference, AUTOTESTCON’06, CA, 2006..

[4] V. Brusoni, L. Console, P. Terenziani, and D. T. Dupre, “A spectrum of definitions for temporal model-based diagnosis,” Artificial Intelligence, vol. 102, no. 1, pp. 39–79, 1998.

[5] O. Contant, S. Lafortune, and D. Teneketzis. Failure diagnosis of discrete event systems: The case of intermittent faults. In Proceedings of the 41st IEEE Conference on Decision and Control, pages 4006–4011, 2002.

[6] A. Correcher, E. Garcia, F. Morant, E. Quiles, and E Blasco-Gimenez, Intermittent failure diagnosis in industrial processes. In IEEE International Symposium on Industrial Electronics, ISIE’03, volume 2, 4006–4011, 2003.

[7] J. de Kleer. Diagnosing intermittent faults. In 18th International Workshop on Principles of Diagnosis, Nashville, USA, 2007.

[8] G. Karsai, J. Sztipanovits, S. Padalkar, and C. Biegl, “Model based intelligent process control for cogenerator plants,” Journal of Parallel and Distributed Systems, vol. 15, pp. 90–103, 1992.

[9] S. Ofsthun and S. Abdelwahed, “Practical applications of timed failure propagation graphs for vehicle diagnosis”, in IEEE Systems Readiness Technology Conference, AUTOTESTCON'07, Baltimore, MD, 2007. (under review)

[10] S. Padalkar, J. Sztipanovits, G. Karsai, N. Miyasaka, and K. C. Okuda, “Real-time fault diagnostics,” IEEE Expert, vol. 6, no. 3, pp. 75–85, 1991.

[11] I. Koren and Z. Kohavi. “Diagnosis of intermittent faults in combinational networks.” IEEE Trans. Computers, 26(11):1154–1158, 1977.

[12] O. Raiman. A circumscribed diagnosis engine. In G. Gottlob and W. Nejdl, editors, Proc. Int. Workshop on Expert Systems in Engineering, pages 90–101. Springer Verlag LNCS 462, 1990.

[13] S. Soldani, M. Combacau, A. Subias, and J. Thomas, Intermittent Fault Diagnosis: a Diagnoser Derived from the Normal Behavior, In 18th International Workshop on Principles of Diagnosis, Nashville, USA, 2007.