Challenges for Model-Based Diagnosis

Ingo Pill
Institute of Software Technology, Graz University of Technology, Austria

Johan de Kleer
c-infinity, Mountain View, CA, USA

Abstract

Since the seminal works by Reiter and de Kleer and Williams published in the late 80's, Model-based Diagnosis has been a significant area of research. This has been motivated by the fact that MBD assists us in tackling a challenge that we face almost on a daily basis, i.e., by MBD allowing us to reason in a structured manner about the root causes for some encountered problem. MBD achieves this in an intuitive, complete and sound way, based on the central idea of investigating the compliance of some observed behavior with a model that describes how a system should behave – given this or that input scenario and parameter set. Over the last 40 years, MBD has been adopted for a multitude of applications, and we saw the emergence of a diverse set of algorithmic, optimizations, as well as extensions to the initial theoretical concepts.We argue that MBD remains highly relevant, with numerous scientific challenges to tackle as we face increasingly complex diagnostic problems. We discuss several such challenges and suggest related topics for PhD theses that have the potential to significantly contribute to the state-of-the-art in MBD research.

2012 ACM Subject Classification Computing methodologies → Causal reasoning and diagnostics

Keywords and phrases Model-based Diagnosis, Diagnosis, Algorithms

Digital Object Identifier 10.4230/OASIcs.DX.2024.6

Acknowledgements We would like to thank the DX community for discussions on topics related to this paper. Among other people, this specifically includes (in a.o.) *Gautam Biswas, Alessandro Cimatti, Alex Feldman, Kai Goebel, Meir Kalech, Ion Matei, Oliver Niggemann, Markus Stumptner, Louise Trave-Massuyes, Franz Wotawa, and Marina Zanella.* This article was influenced also by discussions led at the Dagstuhl Seminar 24031 *Fusing Causality, Reasoning, and Learning for Fault Management and Diagnosis*, where we would like to thank the organizers as well as *Schloss Dagstuhl Leinbniz-Zentrum für Informatik* for organizing and hosting this seminar.

1 Motivation

Model-based Diagnosis (MBD) [51, 11, 12] has been a significant area of research for almost 50 years. It is based on the central idea that diagnosis of a system can be achieved by analyzing how deviations in behavior can be explained by undesired changes in the model of the system. A key advantage of model-based diagnosis is that it relies solely on the formal description of the design of the system, not diagnostic trees, expert systems or human experience. Implementing a concept of reasoning from first principles, we construct the most coherent explanation between our observations on what happened, and our knowledge about how a system should behave. The implicit assumption is that the other parts of a system, i.e., those whose behavior is inconsistent with the information, are those where something went wrong. Being completely flexible in our choice, parts can be system components, measurements, or any other knowledge we have available for exploitation in our diagnostic reasoning process.

MBD is very flexible and thus allows us to focus the diagnostic analysis tailored to the specific context at hand. For a failed regression test, for example, we might want to restrict our focus to those system parts that were most recently changed as well as on components

35th International Conference on Principles of Diagnosis and Resilient Systems (DX 2024).
Editors: Ingo Pill, Avraham Natan, and Franz Wotawa; Article No. 6; pp. 6:1–6:20
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

interacting with them – a technique called program slicing [6] in the context of software engineering. Or we might want to focus only on specific temporal phases in some temporal sequence, like a system's initialization.
In contrast to expert systems, one of MBD's distinct advantages is that if we change or revise individual system parts, we only need to remodel those parts – rather than having to redevelop our entire knowledge base concerning the effects of individual faults and their combinations on the behavior of the entire system.

MBD has proven to be a powerful technique and has been adopted for a variety of problem settings and domains. Nevertheless, there remain open challenges that concern fundamental topics as well as the exploitation of connections to complementing research fields. A prominent example for the latter would be achieving resilience in an autonomous system, which requires diagnostic information about a problem such as to successfully mitigate its effects. In this paper, we articulate a selection of these challenges and encourage researchers to address some of them in their future research.

We will have undoubtedly missed certain challenges to MBD, and we would also like to point out that fellow researchers have similar ideas. So it is not the uniqueness of the challenges that represents the contribution of this paper. Our intention is rather to draw the scientific community's attention to these challenges in order to revive old discussions and trigger new ones on how we can address them in order to propel MBD research to be able to have far wider applicability for the problems of today and tomorrow. We intend to maintain this paper as a living document and will add and modify sections as we learn. So we request the interested reader to send us their comments and inputs on these challenges and we will update this paper accordingly.

2 Preliminaries: a Crash Course in Model-based Diagnosis

While we cover only the very basics, we still aim to provide the interested reader with the relevant background information for our discussions – offering also citations of relevant and more detailed publications.

MBD [51, 12] is a fundamental diagnosis framework that compares information about what happened (in the form of a set of observations OBS) with a description of the system's supposed behavior (in the form of a system description SD). There is no specific required format for expressing OBS or SD, but it includes some sort of reasoning engine that allows us to verify whether OBS lives up to the expectations covered in SD– similar to the task and purpose of a test oracle [46]. Formally, OBS should be consistent with SD, so that $\mathit{OBS} \wedge \mathit{SD}$ is satisfiable.

The innovation of MBD is, not the detection of an inconsistency but that it allows us to *reason about what went wrong* when OBS is inconsistent with SD. That is, implementing a concept of reasoning from first principles, MBD constructs maximum sets of information from SD that are consistent with OBS– implicitly assuming that OBS is correct. Traditionally, MBD algorithms [51, 12, 18, 41, 61] have been designed to be not only sound but also complete, in that they can derive all such maximum sets. The complements (in SD) of these maximum sets define diagnoses. That is, those parts that if they were to deviate from their nominal behavior would explain the observed scenario. There are also incomplete algorithms, i.e., algorithms that might shed some solutions in their search, while guaranteeing, for instance, that they would still find a sound diagnosis of minimal cardinality [14].

Reiter's theory of diagnosis [51] defines the model-based diagnosis of a system as follows (cf. also [12]): A system description SD captures the behavior of a set of interacting components COMP such that SD contains sentences $\neg \mathit{AB}(c_i) \Rightarrow \mathit{NominalBehavior}(c_i)$ encoding a component's behavior under the assumption that it is not operating abnormally. That is, the assumption predicate $AB(c_i)$ triggers a component c_i's abnormal behavior, and $NominalBehavior$ defines its correct behavior in first order logic. Since there are no definitions regarding abnormal behavior, the basic approach is considered to use a *weak fault model* (WFM). Given some actually observed system behavior OBS, a system is recognized to be faulty, if and only if $SD \cup OBS \cup \{\neg AB(c_i) | c_i \in COMP\}$ is inconsistent. A minimal diagnosis for a diagnosis problem (SD, $COMP$, OBS) can then be defined as follows:

▶ **Definition 1.** *A* minimal diagnosis *for (SD, COMP, OBS) is a subset-minimal set* $\Delta \subseteq COMP$ *such that* $SD \cup OBS \cup \{\neg AB(c_i) | c_i \in COMP \setminus \Delta\}$ *is consistent.*

Reiter as well as de Kleer and Williams proposed to compute (all) the minimal diagnoses as the minimal hitting sets of the set of (not necessarily minimal) conflicts for (SD, $COMP$, OBS). This implements the idea that we indeed have to resolve such conflicts that individually express the fact that it is not possible that a specific set of components all work correctly when considering OBS. Thus, a conflict basically describes a set of assumptions that components work correctly that is in conflict with the observations OBS:

▶ **Definition 2.** *A* conflict *CS for (SD, COMP, OBS) is a set* $CS \subseteq COMP$ *such that* $SD \cup OBS \cup \{\neg AB(c_i) | c_i \in CS\}$ *is inconsistent. If no proper subset of CS is a conflict set, CS is a minimal conflict.*

Using a theorem prover or solver capable of returning conflicts, we can compute the conflicts and diagnoses on-the-fly – using a variety of algorithms. Extensions of the first algorithms by de Kleer and Williams [12], Reiter [51], or Greiner et al. [18], are able to avoid redundancies in the computation [41], optimize the memory bloat associated with the open node list [54], and considered the problem from a different point-of-view such as to improve the computation [22]. In addition to new conflict-driven algorithms, researchers have also published *brute force* approaches that compute diagnoses directly in a solver, without explicitly computing conflicts [27, 36]. While such approaches were assumed in Reiter's seminal work [51] to show bad performance, comparisons showed that with today's computation hardware and recent developments in the context of SAT and constraint solvers, they are indeed viable. In particular, it was observed that they seem to offer performance that is at least on par with the original conflict driven approaches [35].

When deploying a *strong fault model* (SFM) MBD approach, we need to define the abnormal behavior – like a stuck-at-1 fault for a logic gate [13]. Depending on the implementation, the abnormal predicates from above would then be replaced by one or more variables that select in which mode a component is supposedly in. Although these additional modeling efforts result in a larger model, they offer the advantage of more specific diagnoses, identifying the actual fault modes of components. Such information is precious insofar as it eases the debugging process significantly by describing the scenario in more details. Furthermore, with specific models – like when diagnosing specifications – a corresponding diagnosis even reports a repair [40] like that we should have used operator F instead of G in a certain place of a formula. As a non-negligible downside, however, the search space for diagnoses grows significantly. That is, with n the number of components/selector variables and m the maximum number of modes for any assumption, it grows from 2^n to $\mathcal{O}(m^n)$.

In our descriptions, we decomposed SD into components, a method commonly used in Model-Based Design (MBD) literature. No such restriction in required. We can use any subset of SD as a block and associate a health state or abnormal predicate with it. Furthermore, MBD is also not limited to using any non-temporal [51, 12] or temporal [40] logic for capturing *SD*. Indeed also other formats, including Petri Nets [9], various kinds of automata [32] or their symbolic implementation can be used (the latter two being close to temporal logics). In particular, we only require the means for checking the consistency of *OBS* with *SD* under the assumptions defined by a diagnosis (and ideally for computing conflicts for the negative case) for being able to employ MBD (see also [51]). We can thus also easily employ SFM MBD using a continuous Modelica model and an oracle that can consider aspects like noisy sensors when checking whether *OBS* matches *SD* for a diagnosis Δ.

Regarding the use of automata for *SD* in a run-time verification context, we would also like to direct interested readers to research on diagnosing discrete event systems There, we often capture faults via unobservable faulty transitions and then investigate means for detecting their presence via inspecting a DES's language and the corresponding observable traces [57]. Such execution-related aspects have been considered also for generating diagnostic hypotheses for hybrid systems [34]. Further, more detailed discussions of MBD, like that considering new observations is non-monotonic, are available in the seminal works [51, 12].

3 Critical Directions for Future DX Research

While MBD has been an active research area since the late 80s, the motivation of being able to formally reason about the root causes for some encountered issue never lost in attractiveness or relevance. The exponential search space and potentially large models entail a certain computational complexity though, so that there is a continuous need to improve on the efficiency of available algorithms, and look for methods to speed up the computations, e.g., by exploiting domain knowledge [56, 42]. Furthermore, there is a constant demand for new MBD engines that would enable us to accommodate new types of models.

In the remainder of our paper, we list of further challenges for MBD. These challenges complement the natural ones as introduced above by focusing on specific questions about certain important aspects of MBD. Before motivating and discussing these challenges in individual subsections, let us compile a brief, comprehensive list first:

- Failure of function, not of a component is the core problem
- The continuous and discrete worlds were never adequately integrated
- Fault propagation
- Wear/component degradation
- Every model is an approximation
- Getting our hands around resilience and how to measure it
- Attacks or failure?
- Improving physical reasoning in MBD
- Exploiting synergies in terms of hypothesis discrimination and transferable insights when having access to multiple system instances
- Parallel computations with multiple cores and hardware acceleration
- Improving diagnostic information provided to human operators and autonomous control
- MBD as a design tool, or “Redesign by MBD”
- Exploiting the full power of integrating symbolic with sub-symbolic computations
- Where does the diagnosis model come from?

While this list is far from being complete, it targets algorithmic, systematic, usability-centered, and other aspects of MBD – which highlights the complexity and diversity of research on MBD. The topics illustrate also how tightly MBD connects and integrates with topics from V&V, debugging, control and system design. Please note that although there are many more topics of interest, we believe that addressing the presented challenges will significantly broaden MBD's applicability to current and future problems.

Each of these challenges merits in-depth discussion in its own dedicated paper, but our intention is to provide an overview. In the following subsections, we thus offer a brief description and motivation for each challenge. We furthermore point out topics for PhD theses that we could imagine to contribute significantly to solving the challenges.

3.1 Failure of Function

When your TV goes black, a repair person will not immediately go to the schematic and analyze what hard- or software component might have changed its behavior (e.g., becoming open or shorted for some electronic part). They will not start by looking at the sound circuit, because they know it cannot cause the black screen. The repair person has and draws on a common-sense intuitive understanding of TVs such that they know the sound circuit cannot cause a problem in the picture circuits. If they find a faulty transistor in the remote control, they will not get distracted by it because even though it is a deviation from the design, it cannot cause the observed symptoms. This topic has received very little attention, yet is fundamental to how humans troubleshoot. This is a very hard research topic because it requires developing a representation of human diagnostic common sense.

From a technical perspective, as humans we intuitively create sort of a dependency graph that captures also failure modes and their effects (and we do this at various abstraction levels), so that we can make quick decisions on a very abstract level without having to consider the entire system in all its detail. Creating technical models that can achieve this, while maintaining completeness of the reasoning process and being fully aware of the potentials and limitations of a specific model, is a severe challenge. Some initial steps have been made, e.g., in the direction of deriving dependency graphs and dependent failure descriptions [59]. The rules/abstract models learned in [29] could be interpreted as very abstract models that describe a systems abstract functionality. A full solution for this challenge could, however, substantially propel research on safety and resilience aspects of autonomous systems, as well as provide important stimuli for research on the efficient and effective control of complex system-of-system architectures.

- Identifying and Learning Abstract Concepts, Dependencies, and Interconnections in a System's Behavior for Their Exploitation in Model-Based Diagnosis
- An Approach to Hierarchical MBD Enabled by Automated Model Abstraction and Refinement Based on Capturing Functional Dependencies
- A Distributed Approach to MBD of Systems-of-Systems Exploiting the Exchange of Abstract System Models For Capturing Functional Interconnections
- Hierarchical and Dynamic MBD for Resilient Resource-Constrained Embedded Systems
- Developing a formal model of human common-sense diagnostic reasoning

3.2 Continuous and Discrete

A typical system has both continuous (e.g., resistor) and discrete (e.g., relay) components. Model-based diagnosis has made significant progress for systems that are purely continuous or purely discrete. Discrete components such as logic gates can be analyzed through causal signal propagation. Continuous systems on the other hand typically do not have direct causal paths. To analyze such systems requires solving simultaneous equations. It is much more difficult to determine causality in such systems because in the worst case every component can cause the observed symptoms. Fortunately, various techniques have been developed to extract some of the causality from continuous systems through techniques such as parameter estimation and structural analysis from the FDI community. However, systems that include both discrete and continuous models remain very challenging.

Such combined models might also surface in scenarios where we use different abstraction levels for describing the individual system parts, e.g., when aiming to optimize our computations. That is, some electronic part might be represented at a logic level, while we need to consider other components at the voltage or energy level for the problem under investigation, even if both are logic gates.

- An Approach to Modeling Systems for MBD that Unites Discrete and Continuous Model Parts and Supports Scaling the Abstraction Level of Individual Parts
- Capturing and Extracting Causality in Analog and Mixed-Signal Designs for Efficient Mode-Based Diagnosis
- Optimizing Model-Based Diagnosis of Mixed-Signal Designs

3.3 Fault Propagation

Fault propagation is rarely modeled in MBD approaches. The output stage of an amplifier may develop an internal short but still function ok. This now overloads the prior amplifier stage. The prior stage now blows its output transistor, causing the overall system to completely fail. MBD will notice the pre-amplifier has failed and repairs it. But it won't take long for it to blow again. An experienced diagnostician would ask themselves the question "Why did the output transistor blow", and understand that the root problem is prior stage not the output stage. They have a model that if a transistor is driving more current than designed, it will fail soon. A simpler version of this issue arises when your home blows a fuse. Sure, replacing a fuse will most probably restore your lights momentarily. But the ultimate cause was a dangling bare wire in your attic.

Previous work like the dependency graphs and dependent failure descriptions computed in [59] made first steps in this direction. With the rising demand for autonomous dependable systems, we however can easily see that we need more elaborate solutions that take details like effects from temporal fault sequences into account, support a continuous refinement during system operation – for example when recognizing that the current model is insufficient – and support resilient autonomous control in hypothesis discrimination [4, 20], fault mitigation [25, 67, 24] and contingency planning [66, 52].

- A Model for Capturing Fault Propagation for MBD Purposes and Deriving Appropriate Inspection and Debugging Strategies
- Enhanced Automated Fault Mitigation in Autonomous Systems by Augmenting MBD With Continuously Refined and Updated Fault Propagation Models
- Integrating Fault Propagation and Prognosis for Exploitation in MBD for Optimizing Resilient Autonomous Control

3.4 Component Degradation

MBD as it was defined originally [51, 12], employs a simple Boolean notion of whether a system's behavior captured in OBS adheres to the nominal one described in SD. In particular, a component is working either correctly or faulty. When incorporating fault modes in our reasoning [13], we're looking at faulty behavior in more detail. That is, in an SFM model we describe how a component can fail and a diagnosis then tells us in which fault mode a component is supposedly in. When incorporating individual fault models, we can thus reason about how a component failed – not how badly, but how. We are, however, not aware of MBD research that considers the positive case in more detail. That is, approaches that assess how good the current status of a system (and its components) is, and its remaining useful life. Currently, and taking aspects like component degradation into account, MBD thus utterly fails to assist us in answering the question: is there some specific component that is likely to fail shortly? The PHM community has various models of remaining useful life (such as utilized in [25]), but none are compatible with MBD. [28] incorporates degradation models in Modelica, but the actual physics based calculation of wear have to be performed by an ad-hoc supplied physics model.

In the context of runtime-verification, a.k.a. monitoring, such considerations have been made. In particular, monitors generated for the Signal Temporal Logic (STL) [16] – which is a logic that merges continuous time and continuous signal values with the basic concepts of Amir Pnueli's Linear Temporal Logic (LTL) [49] – can consider specific quantitative semantics that are in contrast to simple Boolean correct/faulty verdicts like we derive them for MBD. These quantitative semantics allow us thus to capture how well some observed behavior satisfies a property (*SD* in our case), or how badly it failed to do so. To this end, a robustness degree is computed that encodes quantitatively *to which degree* some behavior satisfies or violates a specified STL formula/model. Please note that in run-time verification this means that we investigate a finite prefix that has been observed up to now, and take all its potential continuations into account.

Translating such a concept of quantitative semantics to MBD would be a very interesting extension. This would allow us to judge not only that things are OK, but also the degree to which we should feel safe right now – probably limited to the scope of behavior that is similar to the previously inspected one. It would be only natural to exploit such information in a sensible ranking of diagnoses when deriving inspection strategies based on the diagnoses (complementing the consideration of cardinality and other metrics). For intelligent systems that exploit diagnostic information for achieving operational resilience to disturbances, such quantitative information could also help to assess the severity of issues. This would allow us to extend previous work like [67], where a reliability measure for an agent's action was computed via a data-driven diagnosis approach and then exploited in a monitoring and (re-)planning approach for controlling the agent. Also more elaborate approaches that incorporate MBD and prognosis concepts for achieving operational resilience [25] could profit from this extension to the original MBD theory.

- A Fine-Grained Notion of Consistency for MBD That Supports The Exploitation of Component Degradation
- Resilient Autonomy Supported by Continuous Tracking of Component Degradation via Model-Based Diagnosis

3.5 Models are Approximations

Model-Based Diagnosis is based on having access to an accurate and complete model of the system. If some phenomenon is not modeled, then MBD will have great difficulty pinpointing the ultimate cause for a fault originating from this phenomenon. Let's illustrate this in the case of an overheating resistor R being physically located near a capacitor C on a printed circuit board (PCB) with very minor airflow (like in an enclosure without active cooling). When the heat causes the capacitor C to ultimately fail, MBD will probably pinpoint the capacitor as the culprit – leading to a replacement of the capacitor C but not the resistor R which was the initial root cause. Consequently, R will continue to heat the new C, which will then ultimately fail again (see also our section on fault propagation). If the model had contained a output heat-port for resistor R and an input heat-port for capacitor C, that would only solve one aspect of this challenge. R and C may be very distant from each other on the schematic and only coincidentally be nearby physically, so that the heat model would need to be based on the spatial and physical details of the assembled PCB. In order to address this case, the system model has to include at least: (1) schematic, (2) heat ports on component models, (3) physical distances between actual components.

There are a multitude of scenarios that we can imagine and where the model is limited in its capabilities for the task at hand – due to one or the other reason. Be it a missing (logical or physical) system dependency, some unmodeled environmental input (e.g., some sort of radiation), an inappropriate abstraction layer, a missed change in the environment, or simply a wrong (hidden) assumption. Currently, MBD research offers no concrete means for assessing and expressing our confidence in a model.

Indeed, there is only initial research related to parts of the challenge, like considering novelty in the context of anomaly and fault detection [30], learning models for diagnostic purposes [32] and assessing our confidence in them during the learning process, as well as techniques for diagnosing formal models [40] that can be integrated into workflows for developing models [43]. To the best of our knowledge there is, however, no comprehensive work investigating the challenge as a whole, i.e, which would allow us to continuously assess and express the quality of an MBD model in general, its suitability for a specific diagnosis problem at hand, as well as the confidence we should have in the results computed for a specific diagnosis problem.

In the future, we hope to see MBD evolving such that it offers us the means to assess the individual aspects of a diagnosis model and diagnosis process qualitatively and quantitatively in terms of quality and confidence. This would allow us not only to progress from the intuition that MBD is sound and complete with respect to the model, but it would allow us also to employ further means for verifying the diagnostic hypotheses, i.e., the checks for consistency between *OBS* and *SD*. That is, using a concept of simulation and exploring the space for free parameters, we could come up with a verdict and confidence estimation for this check in situations where the usual mathematically provable yes/no answer is not possible, e.g., due to the lack of a reasoner that is capable of dealing with weak fault models.

- An Investigation of Measures for Assessing the Quality of MBD Models, Their Correlation to MBD Performance in Practice, and Their Application in Hierarchical MBD
- Exploring the Design Space of MBD Models Based on a Quality Measure for Diagnosis Results
- Assessment-Driven Scaling of the Abstraction in MBD Models and its Connection to Fault Mitigation Capabilities in Resilient Autonomous Systems

3.6 Defining Resilience

A resilient system is one that is robust with respect to faults, physical attacks, cyber-attacks, weather and wear. MBD promises to be a key technology to achieve resilience, because it can be embedded in the system itself so that it allows the system to diagnose a situation and respond to disturbances never anticipated by the designer. This type of active resilience provides a dramatic opportunity to improve operational resilience. The MBD DX community has struggled to find a good a definition of resilience, hence it is challenging to measure the improvement in resilience provided by active resilience. Nevertheless, the resilience of factories, systems, supply chains, etc. is an important topic today and MBD can play a key role in achieving it. That is, it provides diagnostic information for creating the awareness that is needed to make the specific connection between component behavior and the correct, desired overall function.

While resilience is not a new topic, its connection to MBD and the development of an integrated theoretical framework have received insufficient attention. At the Dagstuhl Seminar 24031 *Fusing Causality, Reasoning, and Learning for Fault Management and Diagnosis*, there were promising discussions of these aspects (including the formulation of an appropriate notion of resilience) which we hope come to fruition in the near future. Especially in the context of the suggested PhD topics, we would like to pinpoint the interested reader to the opportunities provided by generative MBD as discussed in Sec. 3.13.

- Achieving System Resilience by Exploiting MBD to Generate the Situational Awareness Necessary for Intelligent Control
- A Notion and Measure of Resilience for Continuously Assessing a System's Resilience Capabilities and Performance
- Supporting Operational Resilience at Design Time: Maximize a System's Resilience Capabilities by Adding Appropriate Run-Time Capabilities Based on Autonomous Agency and Model-Based Diagnosis
- An Architecture for a Resilient Agent: Integrating Learning, MBD, Prognosis and Planning for Achieving Resilient Intelligent Control

3.7 Attacks or Failure?

One main motivation of model-based diagnosis is to diagnose the root causes of a failing system, i.e., derive explanations for some observed unexpected behavior that pinpoint to those system parts that are responsible for the problem by exhibiting faulty behavior. When employing MBD, we often make the tacit assumption that it is indeed a fault that is responsible which simply originates from the properties of the natural world. In general, however, the reason for some system part not working as modeled can be many. One such reason is suggested by the hidden MBD assumption that the system description *SD* is per definition correct in terms of the nominal behavior, i.e., we could simply suffer from a fault in the model itself.

As we discussed briefly in Section 3.5, a model and its interaction with the environment are approximations, so that our understanding modeled in *SD* might be incomplete. Or our understanding is complete, but we made a mistake when formally expressing the system dynamics in *SD*. Consequently, we might not experience a system fault leading to a system failure, but suffer from a fault in the model itself that tells us something went wrong (while actually nothing went wrong). When thinking about resilient autonomous systems, such problems can however manifest in undesired system behavior, the useless expenditure of additional resources, or the activation of redundant system parts that contain faults themselves – so that adversarial attacks are feasible.

While such faults can obviously occur also naturally, it is not difficult to fathom a situation where somebody injected a fault on purpose, such as to be able to trigger specific behavior in a certain context. A corresponding adversarial attack could then lead to either passively or actively triggered malignant behavior – an actively triggered scenario requiring the adversary to have the means to actively trigger the injected behavior via providing specific inputs. Such scenarios received considerable attention in image classification [55], for instance. There are multiple options for providing required inputs to a system and implementing appropriate security countermeasures, such as at the sensor level [3]. We would have to consider, however, also scenarios where an attacker simply focuses on a specific sensor at the physical level by blinding parts of a camera's view or by heating up a temperature sensor. While a formal and diagnostic analysis of the model itself might unveil malignant functionality (depending on the modeling style, the detail level, the level of redundancies in the model, as well as the properties we're investigating), identifying such a physical attack could be quite a challenge - depending on the data and techniques available.

Some faults in the model itself could be identified by diagnosing the model, as, e.g., suggested in [40] where the authors diagnose formal properties in the context of example behavior. Tools like RAT, which support the development of formal models through specific workflows, enable us to explore the semantics of models under development [43]. When combined with diagnostic techniques [40], testing methods such as combinatorial testing [46], or technologies developed to address modeling issues (as discussed in Section 3.5), these tools could allow us to identify and explore potential model problems more effectively.

So we could investigate the consistency and integrity in all the available sensor/input information via concepts that investigate either concrete, analytical dependencies, or alternatives that take advantage of virtual sensing methods. The latter, in particular, would employ ML/DL techniques to learn approximations of specific signals from other (sensor) data, which means that we would *learn* approximations of physical and other dependencies between observed quantities rather than having to model them. Either variant would not only enable us to potentially detect faults in sensors, but would provide us also with more context for the task of differentiating between system faults and issues triggered by attacks.

Currently, MBD does not investigate the reasons behind a diagnosis in detail, in that it focuses on whether the provided diagnosis is *consistent* with the observed behavior – but without evaluating the circumstances or their likelihood (at least most of the time). With the ever-increasing system complexity and the importance of autonomy as a central system feature, we argue that will not be enough though. We must consider diagnoses not only as a means to explain a system's failure (in meeting the expectations covered in *SD*), but we need to go one step further and investigate as well as explain diagnoses themselves. This information will provide the necessary background for resilient systems to autonomously react to faults and other issues.

- A Framework for Developing MBD Models With Enhanced Integrity
- Distinguishing Between Faults and Attacks in Model-Based Diagnosis
- Exploiting Virtual Sensing in MBD by Learning Inherent Dependencies for Assessing the Integrity of Diagnostic Computations
- Explaining Diagnoses Resulting from MBD

3.8 Physical Reasoning

Bearings are a weak point in many physical systems. One failure path is that microscopic metal particles released from wear over time get into the races. The race then becomes pitted and wears prematurely. This is in turn increases the clearances between the race and the cage. Eventually the cage winds up carrying some of the load. But the cage is not designed to carry a load, and collapses, producing catastrophic bearing failure and seizure. The prior sentences describe a common-sense model-based approach that people do all the time. But no current MBD approach can perform this type of reasoning.

Human diagnosticians are very good at this type of reasoning. They think through the behavior physical properties of materials, gases, liquids, stresses, heat, etc. For example, the problems with the heat shields on the space shuttle. Modeling all this physics up front is impossible currently. Instead, human diagnosticians bring their physical knowledge to bear when faced with a difficult diagnostic scenario.

Obviously there are tight connections between this challenge and previous ones like, e.g., the challenge of models being approximations only, or those of capturing fault propagation or component degradation. However, the *SD* here is impossible to capture up front, but is elaborated as diagnostic reasoning proceeds. Essentially the appropriate *SD* is constructed as needed from underlying physics of the system being diagnosed.

- Analyzing the Imperfections of MBD Models of Cyber-physical Systems and Their Effects on the Quality of a Diagnosis Process From a Physics-Centered Point-of-view
- Defining and Exploiting A Measure of Confidence in Modeling Physical Processes for Model-based Diagnosis Purposes
- On-Demand Physics-based Common Sense Reasoning for Novel Diagnostic Scenarios
- Introducing Dynamic Abstraction Into the Diagnostic Process

3.9 Synergies

Many systems in our civilization are massively replicated and widely used: Cars, screws, transistors, airplanes, Xerox copiers, etc. It is extremely inefficient for technicians to constantly rediscover the same fault. When a new fault is discovered in one airplane, we would like that knowledge to communicated immediately to the entire fleet. Otherwise the fault will have to be discovered over and over again.

It is not only the reporting of identified design faults that we allows us to profit from synergies. That is, if we debug some faulty scenario autonomously or manually, taking live or stored data from multiple system copies into account might improve the quality of our debugging process. A motivating example would be when we saw that an airplane's autopilot reacted almost always correctly in a certain situation but failed only once, we would intuitively prioritize explanations that are local to the scenario in which it failed. If it failed more than once, we would probably assume a systematic problem in the design, so that we would use the data observed for all airplanes and the problematic scenario to the end of isolating the relevant parameters and mitigating the issue efficiently in a design repair. Relevant applications would be production robots, autonomous and classic cars, drones, planes and many others.

In a scenario with multiple heterogeneous but collaborating agents, synergies generated by inspecting shared data might allow us to distinguish between isolated sensing faults (like when only one agent reports a temperature deviation) and other problems that are spatial to an agent in space or time from issues that concern all agents. The latter might be changes in hidden assumptions, in the environment, or simply unanticipated and thus unmodeled events and dependencies.

An extreme case of illustrating the potential of synergies is a scenario where we would exploit a digital twin (with the obvious challenge of trying to create a scenario as similar to the actual one as possible) for the purpose of being able to actively gather information for hypothesis discrimination. Simulations with the digital twin could allow us to isolate the right diagnosis or at least shrink the solution space significantly (see Sec. 3.11). We see significant potential for exploiting synergies in diagnostic reasoning, particularly in two types of scenarios. First, in autonomous reactive systems with strong safety requirements, such as autonomous transportation in open environments or military applications. Second, in missions with little room for failure due to hazardous environments and high costs of system loss, like space probe applications. Future research on this challenge could investigate it in the context of answering the questions of

- how to do it, i.e., investigating algorithmic options – considering also security and privacy
- when to do it, i.e., assessing the spatial and temporal locality of information, offering an assessment when – and probably also in the context of which diagnosis – to take potential synergies resulting from non-local information into account
- how to adjust local and fleet-wide likelihoods and assess the resulting diagnosis quality
- Approaches to fleet wide diagnostics learning

3.10 Parallelized MBD

Today, even embedded devices like our smartphones feature multiple cores, and mainstream x86 processors from the consumer market from AMD or Intel offer 12 physical cores capable of handling 24 threads[1] at a 250 € price point. When taking also memory prices into account[2], we can easily see that the hardware for exploiting parallel computation has indeed arrived even in low-cost consumer PCs and wearables. Now, while research on distributed diagnosis [50, 63] offer some potential hints at dividing a diagnosis problem into individually solvable chunks, and naive as well as inefficient algorithms for computing multi-scenario diagnoses [48] offer straightforward options for parallelization, it is quite surprising to observe that we have not yet seen much research dedicated to parallel MBD algorithms, with the notable exception of efforts like [21].

Such algorithms could reduce the experienced wall clock time significantly though and thus directly enhance a major aspect of the MBD user experience – by cutting the time we have to wait for diagnoses. For autonomous system, this is a very welcome effect as well, since this would foster much quicker responses to a problematic situation. We thus argue that research on parallel MBD has the potential to significantly boost MBD's utility. If such research would focus furthermore on dynamically scalable approaches that take the currently available cores, power, and memory into account, MBD would be available also to resource-limited autonomous embedded systems which are the backbone of the current age of digitalization and ubiquitous computing.

The most obvious goal would be to investigate options for a multi-threaded approach, i.e., how to leverage the multiple cores for individual tasks arising in MBD computations. That is, we should aim to investigate multi-threading options in terms of single algorithms, multiple stages, and also at the individual task level – like multi-threaded SMT[3] oracles for the consistency check. Complementing these efforts, research on intelligent concepts for sharing information, results, and memory between the threads could significantly optimize the potential (synchronization) overhead introduced with parallel concepts. It would allow us in particular to avoid unnecessary computations (e.g., computing the same conflict more than once) and also loading the same data multiple times from a disk into memory. A non-negligible side effect would be a minimized resource footprint that is highly welcome for reactive and embedded cyber-physical systems like wearables.

Complementing the exploration of parallelization options, it would be very interesting to investigate architectures of dedicated computation hardware for accelerating MBD. For other applications and tasks, e.g., audio and image processing, generic sub-symbolic AI and cryptography, today's processors feature a multitude of special purpose accelerators and architectural choices. We hope that this would be the case in the future also for MBD, potentially exploiting special structured models as suggested in [10].

- Parallelized MBD Computations for Single- and Multi-Scenario Diagnosis Problems
- A Threaded SMT Engine for Diagnostic Purposes
- A Multi-Core Hardware Architecture for Accelerating MBD Computations

[1] AMD Ryzen 5900XT 12 cores/24 threads/4.8 Ghz, `https://geizhals.at`: 15 offers 240-260€ 02/08/24
[2] a 32GB major brand DDR4 3600Mhz DIMM being available at 60-70€, `https://geizhals.at`, 02/08/24
[3] satisfiability modulo theories

3.11 Improving on the Diagnostic Information Provided by MBD

By default, MBD provides us with a set of diagnoses that are sound and complete with respect to the *SD* and *OBS* [51, 12]. Each such diagnosis suggests a set of components (or knowledge blocks) that, if we assume them to be faulty, explain the observed behavior. In the context of debugging, i.e., when we react on the obtained diagnoses, we basically have two options. That is, we either start probing for new data towards the goal of being able to narrow down the number of diagnoses, or we start investigating the components indicated by the diagnoses – ruling out all those diagnoses that contain a component we find to be correct.

Research on sequential diagnosis [53] and probing strategies [20, 19] has been focusing on the former option, i.e., at determining new measurements or distinguishing sequences [4, 62] that should provide us with more and ideally enough information for discerning what actually happened in the system [37]. It must be noted though that, in general, acquiring additional data does not necessarily entail a monotonically decreasing set of diagnoses. Rather, it might be the case that these new data unveils new problems in the system, such that MBD is non-monotonic by default [51, 12].

Irrespective of whether we employ such means for narrowing down the number of diagnoses, we will have to start inspecting the system based on one or more diagnoses at some point. When doing so, we would certainly appreciate any situational data that would accompany the diagnoses. Such data could help in explaining what makes a diagnosis a diagnosis, and would assist us consequently in investigating the component by telling us where to look in the component. This is precious information since the assumption of perfect bug understanding, i.e., the assumption that we can easily isolate the fault in a component, requires extensive expertise with a system and correlates seldomly with reality.

Research on this usability aspect of MBD has been neglected though. This is interesting insofar as the conflicts obtained in the diagnosis process contain such information – like timing information when a fault supposedly occurred in sequential behavior [40]. In MBD algorithms like HS-DAG, GDE or RC-Tree, we exploit, however, only the part of a conflict associated with health state variables/abnormal predicates. The additional explanatory information that was already computed and would be appreciated during debugging, is ignored algorithmically.

One could also imagine to explore further justifications for a conflict/diagnosis (possibly to the extent of offering a complete set) and to provide a user with this valuable diagnostic information that even comes with some guarantees (depending on the exhaustiveness of the exploration) and touches on the ideas behind truth maintenance systems. Research on chronicles and chronicle-based diagnosis [23] considers such justifications, but for serving an entirely different purpose, i.e., that of computing a diagnosis via searching in *OBS* for relevant patterns. We suggest to rather aim at isolating diagnosis problem-specific justifications for the derived diagnoses, and to exploit these data for debugging – and potentially also for deriving probing strategies.

- An Augmented MBD Interface for Enhancing the Effectiveness of Debugging Reactive Stateful Systems
- A Comparison of Augmented User Interfaces for MBD From the Perspective of Automated and Human Debugging

3.12 Where Does the Model Come From?

MBD critically depends on the quality of the model (*SD*) used in the reasoning process. That is, we have to model all the aspects that we want to be taken into account in the reasoning process. As we discussed in Sec. 3.5, a model is always an approximation and it is only as good as we make it (cf. our discussion about wear in Section 3.4, or about attacks and failures in Section 3.7). To this end, we have to aim for an approximation level that allows us to achieve the performance in terms of diagnosability and quality that we're aiming for. In medical diagnosis this relates to epistemic uncertainty (vs. aleatoric uncertainty).

The question is now how we can achieve that, and which type of knowledge we should include in a model. The unsatisfying answer is that it is impossible to provide a universal answer since MBD is completely domain- and task-agnostic from a technological perspective. While this in turn means that we can't come up with universal rules of thumb, we could partly answer this question if there was a notion of confidence in a model. For this we would need to be able to assess a model's quality in the context of the application domain as well as the diagnostic questions we would like to answer.

Aside the consideration of quality aspects, we have take also into account that WFM and SFM models differ to some degree – in that a WFM model leaves a component's behavior completely undefined for the faulty case. Consequently, employing a WFM model does not require us to define all the faulty behavior variants (which we might not be able to do), but it also restricts us in terms of the engines that we can use for the consistency check. For example, we can't use a simulation model for WFM MBD, but it can serve in combination with a test oracle (the latter for assessing the outputs) for SFM MBD if all inputs are known. For SFM MBD, on the other hand, the computations are more complex (see Sec. 2) and we need to obtain and model a complete, representative list of faults and their behavioral descriptions. So there is a trade-off that we need to make between the engines we can use, the information that we require to be available, the efforts we have to spend on the modeling, the preciseness of the results, and the complexity of the reasoning process.

Regardless of a corresponding decision and an application's details, we would certainly appreciate (semi-) automated assistance when we have to create an MBD model. Especially for abductive diagnosis there are indeed concepts available that help a user in generating the appropriate models [44, 45]. With CatIO [33] there is also tool support that allows a user to generate an abductive or an MBD model. As a side-note we would like to point out that SFM MBD models can be indeed very similar or equivalent to abductive models, depending on the abstraction level used and whether we have models of individual components or just encode the effect of some faults on the outputs on a qualitative level.

In addition to workflow assistance for model creation, the challenge of creating MBD models as a symbolic AI technique can greatly benefit from sub-symbolic AI approaches that facilitate model learning. Automata learning [38], for instance, allows us to employ MBD by learning a formal model that can be easily adopted for MBD [32] – supported by tools like AALpy [31]. Virtual sensing concepts (see Sec. 3.7) can be employed to augment existing models with additional signals that approximate certain interconnections and dependencies in a system (or learn aspects from real data that were or could not be modeled). This would support us in recognizing model limitations and issues by comparing the results of different virtual sensors for the same signal or a virtual signal with a real one. There is an abundance of further research on learning models [60, 5, 29] that contributes to tackling some of the aspects of this challenge, but we are to the best of our knowledge not aware of a comprehensive technology, methodology and framework for developing (and assessing) MBD models.

We argue that modeling workflows from other fields, such as those used in developing hardware specifications and models [43], should be adapted for MBD model development. These workflows, and their underlying technologies that help users assess and verify model semantics and quality (as implemented in tools like RAT [8]), could significantly improve the

MBD modeling process. Considering the above mentioned tools, we can, for instance, achieve this easily for models that use LTL-like concepts – like finite state machines or similar logics – due to MBD having been translated already to LTL [40] (sharing even a code-base with technology for generating test oracles [47]).

We do need such technology for a multitude of modeling formalisms though, and we thus require a universal concept and technology base for developing corresponding frameworks. This would allow us to translate technologies like diagnosability assessment [7, 64, 57], design exploration via simulation [43], design assessment via verification [43], diagnostic support during the development [40], and future research published at dedicated conferences like ACM/IEEE MODELS[4], DAC[5] or DATE[6] (in order to name just a view) more easily.

- Formal Analysis of MBD Models for Quality Assessment and Development Support
- A Framework for the Development of MBD Models for Cyber-Physical Systems
- A Formal Approach to Maintaining MBD Models via (Semi-)Automated Correction Triggered by Novelty and Change Detection
- A Framework for a Structured Exploration of the Design Space for MBD Models

3.13 MBD as a design tool, or "Redesign by MBD"

MBD has a long tradition in being used to isolate the root causes of a problem – the purpose for which it was developed. We would like to point out, though, that we could employ MBD also in a generative way. Let us consider that for SFM MBD, we move from answers suggesting that *this set of faulty components explains the problem* to stating that *if these components show this or that alternative behavior, this would match OBS*. As already exploited in [40], a tiny switch in context now unveils that MBD provides us also with the means to suggest repairs – like for the design of a system. That is, we can rethink fault modes as alternative behaviors and then determine a set of changes to a system model (employing alternative behavior for some components) such that *SD* would be consistent again with some supposed witness from a requirements document, i.e., with a trace that describes valid behavior. The same is true for behavior that was described in the requirements document as counterexample, but where we recognize in our design process that it is actually allowed by the system. A similar application domain would that be of planning, i.e., tasks where we seek to reach a desired goal via executing a certain sequence of actions.

This extension to MBD's scope is not entirely new, so that it was exploited for repairing LTL models [40], repairing logical models and interconnections [15] and generating designs via configuring universal components [17]. Technologically, this task is close to solving configuration problems. In practice, it is however the number and complexity of the component's alternative behavior models (which relate to mutation operators as used in mutation testing and fault injection) that determine the effectiveness and performance of a corresponding approach. In [40], for instance, those operators were quite limited in terms of their number and complexity – which is supported by the common (sometimes hidden) assumptions behind the competent programmer hypothesis [2] and an opportunistic reverse view on the coupling effect [39]. These assumptions however not always correlate with reality, and especially so in early design stages. We thus argue that these works were initial steps, and that we need future research on how to efficiently explore the repair/design space in order to fully exploit MBD's generative capabilities.

[4] http://www.modelsconference.org

[5] https://www.dac.com/

[6] https://www.date-conference.com/

It is obvious that the more complex and numerous the mutation operators become, the more optimal a solution we might find – at the cost of a having to conquer an immensely growing search space. For is assessment, we can establish a simple upper bound $O(n^m)$ with m the number of components and n the maximum number of mutation options per component (cf. SFM MBD). For this estimation, we inherently assume mutual exclusiveness of the mutations per component though, and also completely ignore the option of having mutation sequences.

The envisaged problem is thus of very high complexity, and we argue that there is a need for sophisticated solutions that will allow us to conquer the complexity in acceptable time and with acceptable resources. An effective approach would have tremendous potential though and could generate a huge impact. So it would certainly be a most valuable asset for designing, maintaining, and also modeling systems considering the discussion of the previous challenge. The largest impact we see, however, for resilient autonomous reactive systems, i.e., for systems that are allowed to evolve themselves and their behavior in order to adapt to changes in the environment, themselves (degradation), circumstances (regulations), or their mission (like when being required to tackle new tasks). For such systems, we could then add an automated designer to the system that re-designs it at run-time as required for living up dynamic changes as we experience them in an open world.

- Efficient Exploration Strategies for Conquering the Search Space in Generative MBD
- A Generative MBD Framework for Design Space Exploration and Repair
- Achieving Operational Resilience via Adding an MBD-based Run-Time Designer to Reactive Systems

3.14 Fusing Symbolic and Sub-symbolic AI in the context of MBD

In many of the previous sections, we discussed options to fuse symbolic and sub-symbolic techniques for improving MBD. This included sub-symbolic learning of symbolic MBD models, sub-symbolic virtual sensing as a means for assessing the quality of symbolic MBD models, but we could also learn a symbolic model from a neural network to employ MBD for assessing encountered generalization problems [32].

In the literature, we can also find classifiers for diagnosis [26], medical diagnosis based on image classification [65], data-driven approaches like spectrum-based fault localization (SFL) [1] and many other sub-symbolic approaches that complement and compete with MBD. These approaches often allow a much quicker inference/computation of diagnoses compared to MBD, since we move the complexity from the run-time to the training phase, or are content with an approximation as used for SFL. There are also downsides, like that we have to spend a lot of resources when (re-)training classifiers whenever a system changes. The limitations of the trained neural networks are also less graspable from an analytical point of view, i.e., when trying to isolate for which situations we will encounter problems. Furthermore, we have usually little fault data available for training purposes.

As visible from our discussions of the previous challenges, we argue that combining symbolic and sub-symbolic techniques will allow us to leverage the advantages of both approaches in tackling our challenges. While hybrid AI and the fusion of symbolic and sub-symbolic AI concepts are indeed hot topics and are gaining in attention in general, we need to increase our corresponding research efforts also specifically in the field of diagnosis and MBD. Important discussions in this direction were led at the Dagstuhl Seminar 24031 *Fusing Causality, Reasoning, and Learning for Fault Management and Diagnosis*, and we hope to see a lot of upcoming research. Just as diagnosis research in the control community was bridged with diagnosis research in the AI community [58], we need to build similar bridges between the sub-symbolic and symbolic worlds in AI.

- A Survey and Evaluation of Approaches Bridging Symbolic with Sub-Symbolic AI in the Context of MBD

4 Summary

In this manuscript, we showed that Model-Based Diagnosis has never lost in attractiveness, nor in its relevance. There is no doubt that we are still in need of approaches like MBD that allow us to structurally reason about encountered problems, and especially so when taking the ever-increasing complexity of the tasks we are facing in our everyday lives into account. For being able to efficiently and effectively solve the most complex diagnostic tasks, there is, however, an abundance of research questions left that we still need to answer – even after 40 years of research on MBD.

We discussed a corresponding set of MBD challenges that we think are important to address, and which are related to a variety of MBD aspects. We covered the importance and relevance of each challenge in brief discussions, and suggested potential topics for PhD theses that could propel the state-of-the-art in MBD research significantly. While we discussed the challenges individually (hinting at connections every now and then), it is important to point out that there are a lot of cross-connections in terms of motivating agendas – which would in turn multiply the suggested PhD topics from a technological perspective.

In our conclusion, we would like to state again that we were certainly not able to include all the important challenges in this discussion (there is a space limit), and that we would be very grateful for feedback from the interested reader that we could use to improve our discussion for future manuscript versions. As a concluding remark, we would like to express our sincere hopes that this paper would serve as motivation for one or the other researcher and PhD student to work on tackling the described challenges and to contribute to the Model-Based Diagnosis technology of tomorrow.

References

1 R. Abreu, P. Zoeteweij, and A. J. C. van Gemund. On the Accuracy of Spectrum-based Fault Localization. In *Testing: Academic and Industrial Conference Practice and Research Techniques*, pages 89–98, 2007. doi:10.1109/TAIC.PART.2007.13.

2 A. T. Acree, T. A. Budd, R. A. De-Millo, R. J. Lipton, and F. G. Sayward. Mutation Analysis, Technical Report GIT-ICS-79/08. Technical report, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA, 1979.

3 C. M. Ahmed, A. P. Mathur, and M. Ochoa. NoiSense Print: Detecting Data Integrity Attacks on Sensor Measurements Using Hardware-based Fingerprints. *ACM Transactions on Privacy and Security*, 24(1), September 2020. doi:10.1145/3410447.

4 R. Alur, C. Courcoubetis, and M. Yannakakis. Distinguishing tests for nondeterministic and probabilistic machines. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 363–372, 1995. doi:10.1145/225058.225161.

5 J. L. Augustin and O. Niggemann. Graph Structural Residuals: A Learning Approach to Diagnosis, 2023. arXiv:2308.06961, doi:10.48550/arXiv.2308.06961.

6 D. W. Binkley and K. B. Gallagher. Program Slicing. *Advances in Computers*, 43:1–50, 1996. doi:10.1016/S0065-2458(08)60641-5.

7 S. Biswas, D. Sarkar, S. Mukhopadhyay, and A. Patra. Diagnosability Analysis of Real Time Hybrid Systems. In *IEEE Int. Conf. on Industrial Technology*, pages 104–109, 2006.

8 R. Bloem, R. Cavada, I. Pill, M. Roveri, and A. Tchaltsev. RAT: A Tool for the Formal Analysis of Requirements. In *19th Int. Conf. on Computer Aided Verif.*, pages 263–267, 2007.

9 C. Coquand, Y. Pencolé, and A. Subias. Diagnosabilization of Time Petri net for timed fault. *IFAC-PapersOnLine*, 56(2):8648–8653, 2023. 22nd IFAC World Congress.

10 A. Darwiche. Model-based diagnosis using structured system descriptions. *Journal of Artificial Intelligence Research (JAIR)*, 8(1):165–222, June 1998. doi:10.1613/JAIR.462.

11 J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing Diagnoses and Systems. *Artificial Intelligence*, 56(2-3):197–222, 1992. doi:10.1016/0004-3702(92)90027-U.

12 J. de Kleer and B. C. Williams. Reasoning about Multiple Faults. In *5th National Conf. on Artificial Intelligence Volume 1: Science*, pages 132–139, 1986.

13 J. de Kleer and B. C. Williams. Diagnosis with Behavioral Modes. In *11th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1324–1330, 1989.

14 Johan de Kleer. Hitting set algorithms for model-based diagnosis. In *22nd International Workshop on Principles of Diagnosis (DX'11)*, 2011.

15 Johan de Kleer, Alexander Feldman, and Ion Matei. The duality of design and diagnosis. In *29th Int. Workshop on Principles of Diagnosis (DX'18), Warsaw, Poland*, 2018. URL: http://dekleer.org/Publications/ijcai2019r.pdf.

16 A. Donzé, T. Ferrère, and O. Maler. Efficient Robust Monitoring for STL. In N. Sharygina and H. Veith, editors, *Computer Aided Verification*, pages 264–279, 2013.

17 A. Feldman, J. de Kleer, and I. Matei. Design Space Exploration as Quantified Satisfaction, 2021. arXiv:1905.02303.

18 R. Greiner, B. A. Smith, and R. W. Wilkerson. A Correction to the Algorithm in Reiter's Theory of Diagnosis. *Artificial Intelligence*, 41(1):79–88, 1989. doi:10.1016/0004-3702(89)90079-9.

19 B. Han, S.-J. Lee, and H.-T. Yang. Comments on the theory of measurement in diagnosis from first principles. *Information Sciences*, 121(3):349–365, 1999. doi:10.1016/S0020-0255(99)00034-1.

20 A. Hou. A theory of measurement in diagnosis from first principles. *Artificial Intelligence*, 65(2):281–328, February 1994. doi:10.1016/0004-3702(94)90019-1.

21 D. Jannach, T. Schmitz, and K. Shchekotykhin. Parallelized hitting set computation for model-based diagnosis. In *29th AAAI Conf. on Artificial Intelligence*, pages 1503–1510, 2015.

22 U. Junker. QUICKXPLAIN: preferred explanations and relaxations for over-constrained problems. In *19th National Conf. on Artifical Intelligence (AAAI)*, pages 167–172, 2004.

23 X. Le Guillou, M.-O. Cordier, S. Robin, and L. Rozé. Chronicles for On-line Diagnosis of Distributed Systems. In *18th European Conf. on Artificial Intelligence*, pages 194–198, 2008.

24 C. Li, D. Chen, X. Liu, M. Shahidehpour, H. Yang, H. Liu, W. Huang, J. Wang, X. Deng, and Q. Zhang. Fault mitigation mechanism to pave the way to accommodate over 90% renewable energy in electric power systems. *Applied Energy*, 359:122623, 2024.

25 I. Matei, W. Piotrowski, A. Perez, J. de Kleer, J. Tierno, W. Mungovan, and V. Turnewitsch. System Resilience through Health Monitoring and Reconfiguration. *ACM Transactions on Cyber-Physical Systems*, 8(1), January 2024. doi:10.1145/3631612.

26 I. Matei, M. Zhenirovskyy, J. de Kleer, and A. Feldman. Classification-based Diagnosis Using Synthetic Data from Uncertain Models. *Annual Conference of the PHM Society*, 10(1), 2018.

27 A. Metodi, R. Stern, M. Kalech, and M. Codish. Compiling Model-Based Diagnosis to Boolean Satisfaction. In *26th AAAI Conference on Artificial Intelligence*, pages 793–799, 2012.

28 Raj Minhas, Johan De Kleer, Ion Matei, Bhaskar Saha, Bill Janssen, Daniel G Bobrow, and Tolga Kurtoglu. Using fault augmented modelica models for diagnostics. In *Proceedings of the 10th international modelica conference*, pages 437–445, 2014.

29 L. Moddemann, H. Steude, A. Diedrich, I. Pill, and O. Niggemann. Extracting Knowledge using Machine Learning for Anomaly Detection and Root-Cause Diagnosis. In *29th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 2024. to appear.

30 S. Mohan, W. Piotrowski, R. Stern, S. Grover, S. Kim, J. Le, Y. Sher, and J. de Kleer. A domain-independent agent architecture for adaptive operation in evolving open worlds. *Artificial Intelligence*, 334:104–161, 2024.

31 E. Muškardin, B. K. Aichernig, I. Pill, A. Pferscher, and M. Tappler. AALpy: an active automata learning library. *Innovations in Systems and Software Engineering*, 18(3):417–426, 2022. `doi:10.1007/S11334-022-00449-3`.

32 E. Muškardin, I. Pill, M. Tappler, and B. Aichernig. Automata Learning Enabling Model-Based Diagnosis. In *32nd Int. Workshop on Principle of Diagnosis (DX)*, September 2021. URL: `https://www.hsu-hh.de/imb/en/dx-2021`.

33 E. Muškardin, I. Pill, and F. Wotawa. CatIO - A Framework for Model-Based Diagnosis of Cyber-Physical Systems. In D. Helic, G. Leitner, M. Stettinger, A. Felfernig, and Z. W. Raś, editors, *Foundations of Intelligent Systems*, pages 267–276, 2020.

34 S. Narasimhan and G. Biswas. Model-based diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 37(3):348–361, 2007. `doi:10.1109/TSMCA.2007.893487`.

35 I. Nica, I. Pill, T. Quaritsch, and F. Wotawa. The Route to Success - A Performance Comparison of Diagnosis Algorithms. In *23rd International Joint Conference on Artificial Intelligence*, pages 1039–1045, 2013.

36 I.-D. Nica and F. Wotawa. ConDiag - Computing minimal diagnoses using a constraint solver. In *23rd International Workshop on Principles of Diagnosis*, 2012.

37 M. Nica, S. A. Nica, and F. Wotawa. Using Distinguishing Tests to Reduce the Number of Fault Candidates. In *21st International Workshop on the Principles of Diagnosis*, 2010.

38 O. Niggemann, B. Stein, A. Maier, A. Vodenčarevič, and H. Kleine Büning. Learning behavior models for hybrid timed systems. In *26th AAAI Conf. on Art. Intell.*, pages 1083–1090, 2012.

39 A. Offutt. The Coupling Effect: Fact or Fiction? *SIGSOFT Software Engineering Notes*, 14(8):131–140, November 1989. `doi:10.1145/75309.75324`.

40 I. Pill and T. Quaritsch. Behavioral diagnosis of LTL specifications at operator level. In *23rd Int. Joint Conf. on Artificial Intelligence*, pages 1053–1059, 2013.

41 I. Pill and T. Quaritsch. RC-Tree: A variant avoiding all the redundancy in Reiter's minimal hitting set algorithm. In *IEEE Int. Symp. on Software Reliability Engineering Workshops (ISSREW)*, pages 78–84, 2015.

42 I. Pill, T. Quaritsch, and F. Wotawa. Parse tree structure in LTL requirements diagnosis. In *2015 IEEE Int. Symp. on Software Reliability Engineering Workshops*, pages 100–107, 2015.

43 I. Pill, S. Semprini, R. Cavada, M. Roveri, R. Bloem, and A. Cimatti. Formal analysis of hardware requirements. In *43rd Annual Design Automation Conference*, pages 821–826, 2006.

44 I. Pill and F. Wotawa. Fault Detection and Localization Using Modelica and Abductive Reasoning. *Diagnosability, Security and Safety of Hybrid Dynamic and Cyber-Physical Systems*, pages 45–72, 2018. `doi:10.1007/978-3-319-74962-4_3`.

45 I. Pill and F. Wotawa. On Using an I/O Model for Creating an Abductive Diagnosis Model via Combinatorial Exploration, Fault Injection, and Simulation. In *29th International Workshop on Principles of Diagnosis (DX'18)*, 2018.

46 I. Pill and F. Wotawa. Exploiting observations from combinatorial testing for diagnostic reasoning. In *30th Int. Workshop on Principles of Diagnosis*, 2019.

47 I. Pill and F. Wotawa. Extending Automated FLTL Test Oracles with Diagnostic Support. In *IEEE Int. Symp.on Software Reliability Engineering Workshops*, pages 354–361, 2019.

48 I. Pill and F. Wotawa. Computing Multi-Scenario Diagnoses. In *31st International Workshop on Principles of Diagnosis, DX ; Conference date: 26-09-2020*, 2020. URL: `http://dx-2020.org/`.

49 A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57, 1977. `doi:10.1109/SFCS.1977.32`.

50 G. M. Provan. A Model-Based Diagnosis Framework for Distributed Embedded Systems. In D. Fensel, F. Giunchiglia, D. L. McGuinness, and Mary-Anne Williams, editors, *8th Int. Conf. on Principles and Knowledge Representation and Reasoning*, pages 341–352, 2002.

51 R. Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32(1):57–95, 1987. doi:10.1016/0004-3702(87)90062-2.

52 N. Rhinehart, J. He, C. Packer, M. A. Wright, R. McAllister, J. E. Gonzalez, and S. Levine. Contingencies from Observations: Tractable Contingency Planning with Learned Behavior Models, 2021. arXiv:2104.10558.

53 P. Rodler. On Active Learning Strategies for Sequential Diagnosis. In *28th Int. Workshop on Principles of Diagnosis*, volume 4, pages 264–283, 2018.

54 P. Rodler. Memory-limited model-based diagnosis. *Artificial Intelligence*, 305:103681, 2022. doi:10.1016/J.ARTINT.2022.103681.

55 L. Schwinn, R. Raab, A. Nguyen, D. Zanca, and B. Eskofier. Exploring misclassifications of robust neural networks to enhance adversarial attacks. *Applied Intelligence*, 53(17):19843–19859, 2023. doi:10.1007/S10489-023-04532-5.

56 S. Siddiqi and J. Huang. Hierarchical diagnosis of multiple faults. In *20th Int. Joint Conference on Artifical Intelligence*, pages 581–586, 2007.

57 X. Su, M. Zanella, and A. Grastien. Diagnosability of discrete-event systems with uncertain observations. In *25th Int. Joint Conf. on Artificial Intelligence*, pages 1265–1271, 2016.

58 L. Travé-Massuyès. Bridging control and artificial intelligence theories for diagnosis: A survey. *Engineering Applications of Artificial Intelligence*, 27:1–16, 2014. doi:10.1016/J.ENGAPPAI.2013.09.018.

59 J. Weber and F. Wotawa. Diagnosing dependent failures - an extension of consistency-based diagnosis. In *18th Int. Workshop on Principles of Diagnosis (DX-07)*, 2007.

60 A. Windmann, H. Steude, and O. Niggemann. Robustness and Generalization Performance of Deep Learning Models on Cyber-Physical Systems: A Comparative Study, 2023. arXiv:2306.07737, doi:10.48550/arXiv.2306.07737.

61 F. Wotawa. A variant of Reiter's hitting-set algorithm. *Information Processing Letters*, 79(1):45–51, 2001. doi:10.1016/S0020-0190(00)00166-6.

62 F. Wotawa, M. Nica, and B. K. Aichernig. Generating Distinguishing Tests Using the Minion Constraint Solver. In *3rd Int. Conf. on Software Testing, Verification and Validation, Workshops Proceedings*, pages 325–330, 2010. doi:10.1109/ICSTW.2010.11.

63 F. Wotawa and I. Pill. On classification and modeling issues in distributed model-based diagnosis. *AI Communications*, 26(1):133–143, January 2013. doi:10.3233/AIC-2012-0548.

64 L. Ye, P. Dague, D. Longuet, L. B. Briones, and A. Madalinski. Fault manifestability verification for discrete event systems. In *22nd European Conf. on Artificial Intelligence*, pages 1718–1719, 2016. doi:10.3233/978-1-61499-672-9-1718.

65 X. Yu, H. Luo, J. Hu, X. Zhang, Y. Wang, W. Liang, Y. Bei, M. Song, and Z. Feng. Hundredfold Accelerating for Pathological Images Diagnosis and Prognosis through Self-reform Critical Region Focusing. In *33rd Int. Joint Conf. on Artificial Intelligence*, pages 1607–1615, 2024.

66 S. A. Zarghami. Resilience to disruptions: a missing piece of contingency planning in projects. *International Journal of Production Research*, 62(17):6029–6045, 2024. doi:10.1080/00207543.2024.2306474.

67 M. Zimmermann, F. Wotawa, and I. Pill. Pursuing Intelligent Behavior in Cyber-Physical Systems by Lightweight Diagnosis. *Advanced Intelligent Systems*, 4(4), 2022. doi:10.1002/AISY.202100224.