Embedding Diagnosability of Complex Industrial Systems Into the Design Process Using a Model-Based Methodology

Leonardo Barbini[1], Carmen Bratosin[2], and Thomas Nägele[3]

[1,2,3] *ESI (TNO), High Tech Campus 25, Eindhoven, 5656 AE, The Netherlands*
leonardo.barbini@tno.nl
carmen.bratosin@tno.nl
thomas.nagele@tno.nl

Abstract

There is a constant increase of the market expectations on the capabilities of industrial high-tech systems. To meet these expectations, designers of such systems have to explore complex solutions that ensure both functionality and maximum up-time. We describe a methodology that supports the designers in this task. Specifically, we introduce a model-based approach that computes both the diagnosability of a system and the set of hypothetical sensors needed in order to find the root cause of any of the system's failures. The methodology starts at design time, by creating behavioural models for the replaceable parts of the system. These models specify both the expected behaviour and possible Failure Modes (FMs) of the replaceable parts. Using these models, the system design is composed, with the individual replaceable part behaviours defining the system's behaviour. To create these models we use a domain-specific language that generates a Bayesian Network that computes the failure symptoms, i.e., readings on a given sensor configuration, for every FM in the system. Finally, we perform the diagnosability analysis by determining FMs for which the symptoms are equal, causing them to be unidentifiable. For the unidentifiable FMs, we compute a set of hypothetical sensors needed to ensure full diagnosability and the corresponding sensor readings to differentiate between the failures. This information is then used by the designer to make system design trade-offs. We illustrate our approach on two sub-systems of a high-tech machine.

1. Introduction

Designers of today's high-tech industrial systems face the challenging task to deliver simultaneously on two expectations. On the one hand, they have to accommodate the increasing requirements on functionality, so they have to implement designs with higher number of components and more complex control strategies. On the other hand, with the latest trend towards selling industrial assets as a service, they are expected to design systems with an optimal performance and low total cost of ownership.

These two expectations are contradictory: increasing hardware and software complexity inevitably leads to a higher probability of failure, resulting in down-times for which the root-cause has to be diagnosed. The down-times degrade the system's performance and make it overall more expensive to operate. Designers partially mitigate this by deciding for a more robust system, building in hardware redundancy and adaptive control algorithms which compensate for possible components malfunctions. Nevertheless, failures that eventually occur in these systems, made even more complex by such countermeasures, are challenging to diagnose.

To resolve this difficult situation, we propose to provide designers with an overview of the observable effect, at system level, of each of the system's components failures – together with an estimation of the effort required to diagnose. With this system diagnosability overview at hand, designers can explore the design space, for example by adding or removing sensors, and finally choosing between designs with same functionality but different sets of components.

In this paper we implement this proposal by extending the work presented in (Barbini, Bratosin, & van Gerwen, 2020). Specifically, we use diagnostic models generated from a design specification to determine the diagnosability of a system. Furthermore, we compute a list of hypothetical sensors needed to diagnose the system's failures that are not identified through the given sensor configuration. These hypothetical sensors are an indication of either support for the service organization to define diagnostic procedures or placement of additional sensors.

The paper is organised in the following way: Section 2 introduces the relevant literature, Section 3 describes the methodology, Section 4 shows its application on industrial use-cases, Section 5 draws directions for future research and conclusions.

2. Related Research

The literature tackles the task of designing systems that are both complex and reliable from three main angles: procedures for safe system design, computation of the system diagnosability and computation of optimal sensor configurations.

Classical procedures to assess the safety and reliability of systems at design time include: Failure Mode and Effects Analysis (FMEA), Fault Tree Analysis (FTA) and Probabilistic Risk Assessment (PRA) (Mutha, Jensen, Tumer, & Smidts, 2013). In the context of design space exploration as proposed in this paper, the clear limitation of these approaches is that they involve tedious activities. Partially, this can be addressed by using off-the-shelf tools (*MADe*, 2021; Ghoshal et al., 2019) that create such assessments in a structured way.

Diagnosability of industrial systems has been subject of a wide amount of research. (Vignolles, Chanthery, & Ribot, 2020) contains a recent overview. Intuitively, diagnosability is defined as the the ability of a system to exhibit, for each fault situation, different symptoms on the system's monitoring capacities, i.e., the sensors. Examples of diagnosability analysis computed within different diagnostic frameworks are (Provan, 2001) for Model Based Diagnostics (MBD), (Console, Picardi, & Ribando, 2000) for process algebras and (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995) for discrete event systems.

The above examples of diagnosability analysis share three main steps:

- The specification, at component level, of the failure resulting from a given fault.
- The propagation of this behaviour at system-level, from the component towards the sensors, i.e., the computation of a fault signature matrix (Travé-Massuyes, Escobet, & Milne, 2001).
- Classification of the signatures, e.g., diagnosable faults.

The main difference in the characterization of the diagnosability between the diagnostics frameworks consists in the usage of different algorithms to compute the fault signature matrix. Our methodology follows a similar three-step approach, the main difference being in the choice of a Bayesian Network (BN) as reasoning engine. This allows us to deal with lack of observability and provide us algorithms, e.g., message passing inference (Pearl, 1988) and moralized ancestral graphs (Richardson, Spirtes, et al., 2002), to efficiently compute the signatures.

The major difficulty encountered when assessing the diagnosability of a system is an effective way of presenting the results. Usually, this is done by computing the ratio between the number of diagnosable faults over the total number of faults together with reporting the size of sets of faults with identical signatures (Ghoshal et al., 2019). We re-use these classical measurements and propose additional ones based on the number of hypothetical sensors. In this manner, designers have an estimation of the additional work needed for diagnostics.

Sensor placement has been widely investigated as a means to achieve a given level of system diagnosability. This is implemented as an optimisation problem, on all the possible sensor configurations and operational modes for a system, given that the diagnosability can be computed (Scarl, 1994; Daigle, Roychoudhury, & Bregon, 2014). Computational complexity makes this approach unfeasible for large complex systems.

In such cases, instead of the computation of the optimal sensor configuration, the diagnosability analysis has to be complemented with the list of hypothetical sensors needed to diagnose the set of faults with same signatures. In this paper we will adopt this strategy, providing a framework for the generation of such an integrated diagnosability assessment.

3. Methodology

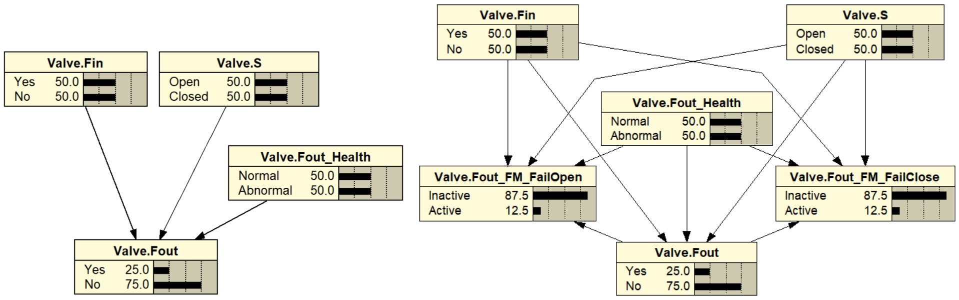
In this paper we use MBD as framework to compute the diagnosability of a system as well as the hypothetical sensors needed to ensure its full diagnosability. This section presents our approach to capture behaviour of failure modes for system's components, together with the relevant algorithms for the computation of the integrated diagnosability analysis.

3.1. Failure Modes in Model Based Diagnostics

MBD defines the diagnostic problem as a consistency check on the tuple (SD, CMP, OBS) (de Kleer & Williams, 1987), where SD is the system description as a finite set of logical formulae, CMP is the set of system components, and OBS is the set of observations. SD contains two types of logical formulae: relations and connections. Relations capture the input-output behaviour of single components, connections capture the interconnections among components, allowing for the propagation of the behaviour from component level to system level.

MBD has a Weak Fault Model (WFM) when only the *normal* behaviour of component c is captured, while the *abnormal* is computed as its negation. MBD has a Strong Fault Model (SFM) (Elimelech, Stern, & Kalech, 2018; Feldman, Provan, & Van Gemund, 2009) when additional information on the *abnormal* is captured, i.e., mapping to specific failure modes. To compute the diagnosability of a system via the fault's symptoms using MBD, the model must have a SFM.

In (Barbini et al., 2020), we translate MBD to a probabilistic reasoning problem where we use BN as inference engine. A BN (Pearl, 1988) is a directed acyclic graph representation of



(a) BN using a WFM.

(b) BN using a SFM.

Inputs		Output	Health	Failure Mode
F_{in}	S	F_{out}		
Yes	Open	Yes	Normal	
Yes	Closed	No	Normal	
No	Open	No	Normal	
No	Closed	No	Normal	
Yes	Open	No	Abnormal	FailOpen
Yes	Closed	Yes	Abnormal	FailClose
No	Open	Yes	Impossible	
No	Closed	Yes	Impossible	

(c) Mapping of behaviours.

Figure 1. An example model of a valve.

the joint probability distribution $P(V)$ over a set of random variables V. The arcs of the graph express causal dependence relationships between the variables. For each of the nodes is defined a local conditional probability table, i.e., $P(n \mid I)$ where $I \subseteq V \setminus \{n\}$ is a set of input nodes for node $n \in V$.

Let us consider component $c \in CMP$ represented by its inputs I and its outputs O. We translate the logical formulae of SD, into conditional probabilities of O on I. Furthermore, we add *Health* nodes H with states *Normal* and *Abnormal* to every inputs/output relation as in (Barbini et al., 2020). For a component c its definition as BN becomes:

$$P(c) = P(I, O, H) = P(I) \cdot P(H) \cdot P(O \mid I, H) \qquad (1)$$

Note that H is independent of I, similar to (Flesch, 2008) and that a component has one health node per output node.

Since the faulty behaviour is modeled by means of health node H, this model is a Horn Strong-Fault Model (HSFM), as defined by (Feldman et al., 2009). Figure 1a shows the BN for a valve using this type of model[1].

We extend the approach above by adding a more refined description of the faults. We limit this faulty model description to the negation of the *normal*, i.e., a negative literal Strong-Fault Model (nlSFM) (Feldman et al., 2009).

One might extend to nlSFM, by setting the states for H to the union of *Normal* and *Faults*, such as in (Borth & Barbini, 2019), thus occurrence of a fault is independent of the component's inputs. However, design engineers find it more practical to describe failure modes, i.e., the manifestation of a fault in the failed behaviour of a component. In this case, a failure mode is dependent on both the component's inputs and outputs.

Let's consider the example of a valve. A valve has as inputs the flow of the water (F_{in}) and the desired state of the valve (S), and as output the water flow (F_{out}). The valve's faults manifest via failure modes: *FailOpen* and *FailClose*, of which the first indicates that the $F_{out} = 0$ when $S = Open$, and the second $F_{out} \neq 0$ when $S = Close$. These two failure modes depend on input S.

To accommodate the above behaviour, we add failure modes to the BN as the set of nodes FM such that:

$$P(I, O, H, FM) = P(I, O, H) \cdot P(FM \mid I, O, H) \qquad (2)$$

Each failure mode node has two states: *Active* and *Inactive*.

Figure 1b shows the BN for the valve model according to the above formulae and Figure 1c presents the $P(I, O, H, FM)$. Note that the last two rows are mapped on *Impossible*, indicating that those combinations of I and O have $P(F_{in} = No, F_{out} = Yes) = 0$.

For the model defined above, a diagnosis D represents the assignment of the health nodes H to *Normal* or *Abnormal* together with the assignment of failures modes FM to *Active* or *Inactive*, accordingly.

3.2. Computation of Diagnosability

Diagnosability (Vignolles et al., 2020) represents the ability of a system to identify failures via their symptoms. In this paper, we compute the diagnosability of a system by analysing the observability of FM.

To compute the observability of FM for a given system, its operational context and monitoring capacity must be defined. The operational context is represented by the system's inputs that are independent of the defined behaviour, such as configuration parameters or connection of the modeled system to other systems such as power inputs. The monitoring capacity are all the properties readable from the system that, as an ensemble, define a failure signature.

In our BN translation of a system specification, the operational context is given by a state assignment for all input nodes that are root nodes. We will refer to this as the base evidence BE of the BN. The monitoring capacity is a subset of the output nodes O and is specified as a set of sensor nodes by the designers. We will refer to this subset as the sensor configuration SC, with $SC \subseteq O$.

[1] For visualization of the BN we use Netica from (*Norsys*, 2019)

Algorithm 1 Fault signature matrix computation algorithm.

```
function ComputeFSM(CMP,BE,SC)
    $FSM \leftarrow$ InitializeMatrix()
    $FM \leftarrow$ FailureModeNodes($CMP$)
    $H \leftarrow$ HealthNodes($CMP$)
    for $f \in FM$ do
        $D \leftarrow \{f = Active\}$
        $H' \leftarrow H \setminus$ RelatedHealthNode($f$)
        $D \leftarrow D \cup \{h = Normal \mid h \in H'\}$
        $sig(f) \leftarrow$ ComputeSignatures($SC, BE, D$)
        $FSM \leftarrow$ Append($FSM, [f, sig(f)]$)
    end for
    return $FSM$
end function
```

3.2.1. Impact Analysis of Failures

To analyse the diagnosability of a system, we use the MBD model as defined in Section 3.1, to simulate the effect of every single failure on SC, i.e., the failure signature. This results in a matrix that provides the signature of every failure: the Fault Signature Matrix (FSM). The computation of the FSM given CMP, BE and SC is shown in Algorithm 1.

Algorithm 1 first collects all failure modes FM and health nodes H from the components CMP. Then a diagnosis D is constructed for every failure mode $f \in FM$. To construct D, f is set to *Active* and all health nodes unrelated to this failure (H') are set to *Normal*, as this will ensure that all other failure modes are *Not Active*. Thus, the algorithm considers only diagnoses with a single failure at the time, but can easily be extended to support multiple failures.

Function `ComputeSignatures` computes the set $sig(f)$ of all possible signatures of f on SC, given BE and D as:

$$sig(f) = \{sig \in \prod_{s \in SC} \Sigma(s) \mid P(sig \mid BE, D) > 0\} \quad (3)$$

where $\Sigma(n)$ is the set of all state assignments for node n. Specifically, $sig(f)$ is given by the elements of the Cartesian product over $\Sigma(s)$ for $s \in SC$ that have a positive probability given the diagnosis D and base evidence BE.

Finally, all the signatures are added to the FSM. Note that there may be multiple signatures possible for some failure modes, causing them to occur multiple times in the resulting FSM.

3.2.2. Observability and Service Actions

While the FSM contains failure signatures on the sensors in sensor configuration SC for all failures, it does not yet provide insight in how diagnosable the system is. Some failures may have identical signatures, which makes them indistinguishable from each other via SC. The more failures have different signatures, i.e., diagnosable failures, the better the diagnosability of the system. To have a system with full diagnosability, it typically must be fully observable, i.e., all the system's components' outputs are in SC.

Since this is not feasible for most systems, we developed Algorithm 2 to compute an extended FSM ($extFSM$) based on the set of hypothetical sensors ($hpSC$), i.e., additional measurements, needed to discriminate between failure modes having identical signatures on SC, for all non-distinguishable failure modes. These measurements correspond to output nodes distinguished from SC, i.e., $hpSC \subseteq O \setminus SC$.

Let us first introduce a set of definitions, similar to (Travé-Massuyes et al., 2001), that will be used in the remaining of the paper.

Definition 3.1 *Normal behaviour of a system $sig(Normal)$ is the set of signatures on SC given BE and assuming that all components are Normal:*

$$sig(Normal) = \{sig \in \prod_{s \in SC} \Sigma(s) \mid P(sig \mid BE, D_N) > 0\}$$

where $D_N = \{h = Normal \mid h \in H\}$.

Definition 3.2 *A failure mode $f \in FM$ is called diagnosable for signature $sig \in sig(f)$ if and only if signature sig is unique, i.e., $\forall g \in FM \setminus \{f\}$ $sig \notin sig(g)$, and $sig \notin sig(Normal)$, i.e., its signature discriminates from normal behaviour.*

Definition 3.3 *A failure mode $f \in FM$ is called D-class diagnosable for signature $sig \in sig(f)$ if and only if signature sig is not discriminable, i.e., $\exists g \in FM \setminus \{f\}$ such that $sig \in sig(g)$.*

Definition 3.4 *A failure mode $f \in FM$ is not observable for signature $sig \in sig(f)$ if and only if $sig \in sig(Normal)$, i.e., signature sig is identical to a signature obtained for normal behaviour.*

Note that a failure mode with multiple signatures can be at the same time not observable and (D-class) diagnosable, for different signatures. For this reason, in Algorithm 2, we consider failure mode-signature pairs (f, sig).

Algorithm 2 returns the set of diagnosable failure modes FM_D, failure modes that are not observable FM_{ND} and the set of extended FSM for all D-class failures $DFSM$.

The algorithm starts by computing the FSM as introduced above. Then failure modes are grouped by their signatures in FMG via helper function `GroupFMbySig`. Specifically, FMG is the set of tuples (fm, sig), where $fm \subseteq FM$ is the set of failure modes for which every $f \in fm$ has the same signature $sig \in sig(f)$.

For every tuple in FMG, Algorithm 2 discriminates between the set of not observable failures FM_{ND}, the set of diag-

Algorithm 2 Computation of $DFSM$, FM_D and FM_{ND}.

```
function ComputeExtendedFSM(CMP,BE,SC)
    $DFSM \leftarrow \{\}$, $FM_D \leftarrow \{\}$, $FM_{ND} \leftarrow \{\}$
    $FSM \leftarrow$ ComputeFSM($CMP, BE, SC$)
    $FMG \leftarrow$ GroupFMbySig($FSM$)
    for $fm, sig \in FMG$ do
        if $sig \in sig(Normal)$ then
            $FM_{ND} \leftarrow FM_{ND} \cup \{(fm, sig)\}$
        else if $|fm| == 1$ then
            $FM_D \leftarrow FM_D \cup \{(fm, sig)\}$
        else
            $s \leftarrow$ ReachedNodes($CMP, BE, fm$)
            $hpSC \leftarrow s \setminus SC$
            $E \leftarrow BE \cup sig$
            $CMP' \leftarrow$ RestrictToFM($CMP, fm$)
            $hpFSM \leftarrow$ ComputeFSM($CMP', E, hpSC$)
            $extFSM \leftarrow (fm, sig, hpSC, hpFSM)$
            $DFSM \leftarrow DFSM \cup \{extFSM\}$
        end if
    end for
    return $FM_D, FM_{ND}, DFSM$
end function
```

nosable failure modes FM_D and D-class diagnosable failure modes.

Then for each D-class, i.e., failure modes that have the same signature, the function `ReachedNodes` computes the set of output nodes affected by the failure modes with the same signature. To compute this, all dependent output nodes for every $f \in fm$ are computed using a moralized ancestral graph (Richardson et al., 2002) of the BN. Specifically, this method indicates the output nodes that are affected when a failure mode node is set to $Active$. All affected output nodes that are not yet in SC are stored in $hpSC$, as these are hypothetical sensors.

Subsequently the function `RestrictToFM` restricts the failure modes in CMP to only those in fm such that $CMP' = (I, O, H, fm)$. Then, a new fault signature matrix $hpFSM$, is computed for these hypothetical sensors, given BE together with sig. Finally, the extended FSM $extFSM$ is constructed as a tuple containing the following four elements:

- The failure modes covered by the service action (fm).
- The signature of fm on SC (sig).
- The output nodes to measure for discriminating between the failure modes ($hpSC$). Note that $hpSC \cap SC = \emptyset$.
- The FSM for fm on sensors in $hpSC$ ($hpFSM$).

The set of all extended FSM, i.e., $DFSM$, gives designers insight into potential placement for additional sensors given the provided context BE. Simulations for different operational contexts could be executed to compare results and come to a final SC. When SC is definitive, the service organization can use the information to create effective diagnostic procedures by using $hpFSM$ as a decision matrix for manual intervention.

3.3. Diagnosability Metrics

To support the designers in the design space exploration, the results of the diagnosability analysis should be presented effectively. A common way to do this is to summarize the results in diagnosability metrics.

A classical metric (Ghoshal et al., 2019) is RD, the ratio of diagnosable FM over the number of possible FM signatures in a system:

$$RD = \frac{|FM_D|}{|FSM|} \tag{4}$$

We complement this metric with additional ones describing the complexity of the $DFSM$. Specifically, we compute the average number of failure modes in $DFSM$ for each D-class:

$$\overline{fm} = \frac{1}{|DFSM|} \sum_{DFSM} |fm| \tag{5}$$

together with the average number of hypothetical sensors in $DFSM$ for each D-class:

$$\overline{hpSC} = \frac{1}{|DFSM|} \sum_{DFSM} |hpSC| \tag{6}$$

where $DFSM$ is the set of tuples $(fm, sig, hpSC, hpFSM)$ as in Algorithm 2. We also estimate the maximum number of hypothetical sensors MHS, as an indication of the worst case diagnostic scenario:

$$MHS = \max_{|hpSC|} DFSM \tag{7}$$

Note that for a fully diagnosable system $|FM_D| = |FSM|$ and $|DFSM| = 0$, therefore $RD = 1$.

Often industrial systems are not fully diagnosable and the target values for the diagnosability metrics are decided by experts. In the next section, we use these metrics to assess the diagnosability of two industrial subsystems and discuss the insights obtained.

4. Applications

This section describes the application of the MBD diagnosability analysis of Section 3 on several use cases. We first present our methodology applied on a sample system on which we perform design space exploration, then we present the results obtained on two industrial subsystems.

All the system's models are constructed using a Domain Specific Language (DSL) as in (Barbini et al., 2020). The DSL uses an object-oriented paradigm, starting with the creation of classes for the components and then specifying the system, by creating instances of these components and interconnecting them together, according to the given design specification. This ensures a fast model creation as confirmed by design engineers. For all the computations on the BN of Algorithms 1 and 2, we use the aGrUM (a Graphical Universal Modeler) library (Gonzales, Torti, & Wuillemin, 2017).

4.1. Small System Example

This section focuses on the application of the methodology on a small hydraulic system. Figure 2 shows the schematic of the system. In sequence we have a valve, a pipe, a heater and two sensors to measure flow and temperature of the water respectively. Water flows from the valve towards the heater. Note that we do not model a pipe from the output of the heater to the sensors.

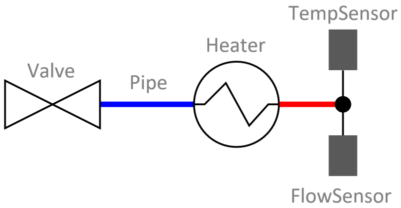


Figure 2. Schematics of a small hydraulic system.

Table 1 lists the system's components together with their inputs, outputs and failure modes. We consider two types of inputs and outputs, flow *F* and temperature *T*, both having three possible states *Low, Normal*, and *High*. Additionally, *S* represents the state of the valve, *Open* or *Closed*; *P* represents the heater power, *On* or *Off*. Note that sensor readings have an extra state *Improbable* signifying a reading that is out of the normal range of the sensor.

To compute the diagnosability of the system we specify SC and BE, i.e., monitoring capacity and operational context. SC is given by *FlowSensor.*F_r and *TempSensor.*T_r. Here the notation $c.n$ stands for node n in component c. BE is chosen as the state assignment for the system's input nodes: $\{$*Valve.*F_{in} = *Normal*, *Valve.*T_{in} = *Normal*, S = *Open*, P = *On*$\}$. This context assumes that all systems connected to this example system are operating correctly.

The FSM for the system is presented in the first three columns of Table 2. The last column maps the failure mode signature to the set of diagnosable failures FM_D or to a specific D-class (DC$_n$). Note that several failure modes appear multiple times in the table, due to their manifestation via different signatures. Furthermore, the failure mode *Valve FailClose* does not appear because it cannot manifest given the chosen BE.

Table 1. Inputs, outputs and failure modes of the components in the small hydraulic system.

Component	Inputs	Outputs	Failure Modes
Valve	F_{in} T_{in} S	F_{out} T_{out}	FailClose FailOpen
Pipe	F_{in} T_{in}	F_{out} T_{out}	Leaking
Heater	F_{in} T_{in} P	F_{out} T_{out}	NotHeating Overheating
FlowSensor	F_{in}	F_r	Broken
TempSensor	T_{in}	T_r	Broken

Table 2. FSM for the small hydraulic system together with mapping to FM_D and D-class.

Failure Mode	T_r	F_r	
FlowSensor Broken	Normal	Improbable	FM_D
FlowSensor Broken	Normal	NoFlow	FM_D
TempSensor Broken	Improbable	Normal	FM_D
Heater Overheating	High	Normal	DC_1
TempSensor Broken	High	Normal	DC_1
Evidence T_{in} mismatch	High	Normal	DC_1
Pipe Leaking	Low	NoFlow	DC_2
Valve FailOpen	Low	NoFlow	DC_2
Evidence S mismatch	Low	NoFlow	DC_2
Evidence F_{in} mismatch	Low	NoFlow	DC_2
Evidence P mismatch	Low	Normal	DC_3
Heater NotHeating	Low	Normal	DC_3
TempSensor Broken	Low	Normal	DC_3
Evidence T_{in} mismatch	Low	Normal	DC_3
FlowSensor Broken	Normal	High	DC_4
Evidence F_{in} mismatch	Normal	High	DC_4
FlowSensor Broken	Normal	Low	DC_5
Pipe Leaking	Normal	Low	DC_5
Valve FailOpen	Normal	Low	DC_5
Evidence F_{in} mismatch	Normal	Low	DC_5

The failure mode entries such as *Evidence n mismatch* capture the possibility that node n could have a different state than what is assumed in BE. Such a mismatch could explain certain abnormal behaviour captured by the sensors. We do so, to capture also the cases when the failure is outside of the boundary of the system modeled (e.g., a user error).

Table 3 presents the hypothetical FSM for D-class DC$_5$. The first column lists the failure modes corresponding to DC$_5$ in Table 2, the other columns indicate the hypothetical sensors $hpSC$ needed to diagnose the failure modes. Each cell entry lists the signature for the given failure mode and $hpSC$ combination. $hpSC$ correspond to readings of the outputs of system's components not already present in SC, i.e., state assignment of the corresponding nodes in the BN. Table 3 extends the failure modes signature on the SC from Table 2, i.e., $\{T_r$ = *Normal*, F_r = *Low*$\}$, with the $hpSC$ signature.

Table 3. Example of hypothetical FSM for D-class DC$_5$.

	Hypothetical sensors			
Failure Mode	*Pipe.*F_{out}	*Heater.*F_{out}	*Valve.*F_{out}	F_{in}
FlowSensor Broken	Normal	Normal	Normal	Normal
Pipe Leaking	Low	Low	Normal	Normal
Valve FailOpen	Low	Low	Low	Normal
Evidence F_{in} mismatch	Low	Low	Low	Low

Design Space Exploration

The diagnosability analysis presented above assumes that water flow and water temperature are measured at the outlet of the small system. During design space exploration, the designer might consider placing an additional sensor or moving one of the two already considered to a different location.

The metrics in Equations 4-7 are computed to compare the diagnosability for different SCs. Table 4 lists the metrics together with the size of not observable failure modes and the size of the sensor configurations. SC_1 is the sensor configuration in Figure 2, and is our zero-measurement.

In Table 3, we notice that by measuring *Valve.*F_{out} we can reduce the table size to half. We create a new sensor configuration SC_2 by adding a flow sensor positioned at the outlet of the valve. As expected, adding a sensor greatly improves the RD, while partially improving $\overline{fm}$ and $\overline{hpSC}$. However, it does not decrease the maximum number of hypothetical sensors that need to be introduced to identify the failure (MHS).

We also analyze sensor configuration SC_3, which has a flow sensor at the outlet of the valve and a temperature sensor at the outlet of the heater. SC_3 allows for the unique identification of faults in the valve, i.e., higher RD and lower $\overline{fm}$, but does not allow identification of *Pipe Leaking*, as captured with FM_{ND} of size 1.

Table 4. Diagnosability metrics for small system example.

Metric	$\mathbf{\|SC_1\| = 2}$	$\|\mathbf{SC_2}\| = 3$	$\|\mathbf{SC_3}\| = 2$
RD	0.15	0.42	0.2
$\overline{fm}$	3.40	2.80	3.2
$\overline{hpSC}$	2.40	1.80	2.2
MHS	3.00	3.00	4
$\|FM_{ND}\|$	0	0	1

Combined with knowledge on costs of additional sensors and diagnosability targets, such comparisons support the system designer in making decisions regarding sensor placement. Note that this is a complex decision process and even in this example it is not trivial to choose the best sensor configuration.

4.2. Industrial Applications

This section describes two industrial system to which we applied the methodology. Due to confidentiality, the use case descriptions are limited to a short summary and their diagnosability figures.

4.2.1. Hydraulic System

The first industrial application concerns a hydraulic module of a lithography machine. This hydraulic system thermally conditions water and supplies it to several modules within the machine. Water has to be supplied both at correct temperature and flow.

The hydraulic components in the system are: pipes, valves, manifolds and flow silencers. The water temperature is conditioned via cartridge heaters. The system has water flow and temperature sensors, together with differential pressure sensors and switches, both for pressure and temperature. The latter ensure safety by switching off the water flow or power supply to the heaters when outside the required range. The system has a total of 263 components, of which 77 are sensors.

The BN for this system contains more than 2000 nodes of which 400 are failure modes. For the diagnosability analysis we consider that the water is supplied to the inlets of the system at the correct flow and temperature, in spec with the system's heating capacity. Further, all the valves are set to open, as these will only be closed while servicing the module. Note that our methodology accounts for valves erroneously left closed after service, via the *Evidence n mismatch* as in Table 2.

The SC contains all the output nodes for the 77 sensors together with software logging from the supplied modules, for a total of 104 nodes. The diagnosability metrics for the hydraulic system are presented in Table 5. According to the system's designers, given the complexity the figure for diagnosable failure modes is acceptable and the metrics for the hypothetical sensors are low enough. Further, the failure modes covered by the service action matrix related to MHS have low impact on the system's functionality, i.e., not affecting supplied modules but limited to components in the return part of the hydraulic circuit.

The designers validated the approach by inspection of the FSM and $DFSM$. They acknowledged that the methodology provides them with evidence, otherwise difficult to obtain at design time, on the efficiency of their diagnostic system. Additionally, they investigate the usage of our the model to facilitate the creation of reliability deliverables such as FMEA. This is estimated to save them more than 200 man-hours.

4.2.2. Heating System

The second industrial system to which we applied the methodology is a heating system. The heating system is a conditioning module that consists of a set of heaters to maintain correct temperatures. Included in the system are controlled power supplies, power distribution boxes, power cables, heaters and

Table 5. Diagnosability metrics for the industrial systems.

Metric	**Hydraulic**	**Heating**
RD	0.49	0.072
$\overline{fm}$	2.86	3.26
$\overline{hpSC}$	2.00	2.51
MHS	9	5
$\|FM_{ND}\|$	0	4

temperature sensors. In total, the system contains more than 500 replaceable parts, of which 200 are sensors.

The BN contains more than 6500 nodes, including 900 failure mode nodes and 150 nodes that provide input to the system. Most system inputs cover set-points for power supplies and safety switches that could cut power in specific parts of the system. To perform the diagnosability analysis on the system, the chosen BE expects all set-points to be *On* and all safety switches to be *On*, i.e., all parts of the system are powered. The SC contains the temperature sensors (80) and power monitoring sensors (120), i.e. voltage and current measurements. Additionally, the set-points as logged by the software are also included in the SC. Altogether, SC contains 240 nodes.

Table 5 shows the diagnosability metrics for the heating system. We validated the model by comparing with available diagnostics procedures for the previous design. Our model covered all the failure modes presented in the procedures. The added value was recognized in the fact that our model, by computing failure signatures on SC at system level, enables service engineers to reduce the size of their procedures, and thus decrease diagnostic time.

5. Conclusions

In this paper we showed how applying Model-Based Diagnostics (MBD) at design time gives valuable insights into the diagnosability of a system. Additionally, we presented the usage of MBD in the creation of input for the service organization – by computing the possible hypothetical sensors required to distinguish between failures that present identical signatures.

The proposed approach provides insight in the diagnosability of the system that would otherwise have been difficult to obtain. The results of the diagnosability analysis are used to generate reliability deliverables comparable to those currently manually created, thus saving time during the design process. The model-based approach ensures completeness of the deliverables. Finally, the method allows for comparison of different sensor deployment strategies. These conclusions were acknowledged by the design-engineers of the industrial systems for which the methodology was applied.

Currently, for the models created, the method proved scalable. However, the size of the model quickly increases when multiple different subsystems are integrated in one model. A hierarchical approach to perform the computations on the model could prove beneficial. At this moment, we experiment with available methods to understand their limitations and applicability to our methodology.

Furthermore, since our models provide feedback only for a specific instance in time, we investigate how integrating this discrete event systems based diagnostic approach will allow us to perform diagnostics over time and capture dynamic behaviour of control system.

Acknowledgement

The research is carried out as part of the ADiA program under the responsibility of ESI (TNO) with ASML as the carrying industrial partner. The ADiA research is supported by the Netherlands Ministry of Economic Affairs.

References

Barbini, L., Bratosin, C., & van Gerwen, E. (2020). Model based diagnosis in complex industrial systems: a methodology. In *PHM Society European Conference* (Vol. 5, pp. 8–8).

Borth, M., & Barbini, L. (2019). Probabilistic health and mission readiness assessment at system-level. In *Proceedings of the Annual Conference of the PHM Society* (Vol. 11).

Console, L., Picardi, C., & Ribando, M. (2000). Diagnosis and diagnosability analysis using process algebra. In *Proceedings of the Eleventh International Workshop on Principles of Diagnosis (DX-00), MX, Morelia, Mexico* (pp. 25–32).

Daigle, M., Roychoudhury, I., & Bregon, A. (2014). Diagnosability-based sensor placement through structural model decomposition. In *Second European Conference of the PHM Society 2014* (pp. 33–46).

Elimelech, O., Stern, R., & Kalech, M. (2018). Structural abstraction for model-based diagnosis with a strong fault model. *Knowledge-Based Systems*, *161*, 357–374.

Feldman, A., Provan, G., & Van Gemund, A. (2009). Solving strong-fault diagnostic models by model relaxation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (p. 785–790). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Flesch, I. (2008). *On the use of independence relations in bayesian networks*. Radboud University, Nijmegen.

Ghoshal, S., Deb, S., Haste, D., Hess, A., Zahiri, F., & Sutton, G. (2019). An Integrated model-based Approach for FMECA Development for Smart Manufacturing Applications. In *Annual Conference of the PHM Society* (Vol. 11).

Gonzales, C., Torti, L., & Wuillemin, P.-H. (2017, June). aGrUM: a Graphical Universal Model framework. In *International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems*. Arras, France.

Made. (2021). `https://www.phmtechnology.com/`.

Mutha, C., Jensen, D., Tumer, I., & Smidts, C. (2013). An integrated multidomain functional failure and propagation analysis approach for safe system design. *Artificial Intelligence for Engineering Design, Analysis and*

Manufacturing: AI EDAM, *27*(4), 317.

de Kleer, J., & Williams, B. C. (1987). Diagnosing multiple faults. *Artificial Intelligence*, *32*(1), 97 - 130.

Norsys. (2019). `www.norsys.com/netica.html`.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Provan, G. (2001). System diagnosability analysis using model-based diagnosis tools. In *Component and systems diagnostics, prognosis, and health management* (Vol. 4389, pp. 93–101).

Richardson, T., Spirtes, P., et al. (2002). Ancestral graph markov models. *The Annals of Statistics*, *30*(4), 962–1030.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on automatic control*, *40*(9), 1555–1575.

Scarl, E. (1994). Sensor placement for diagnosability. *Annals of Mathematics and Artificial Intelligence*, *11*(1), 493–509.

Travé-Massuyes, L., Escobet, T., & Milne, R. (2001). Model-based diagnosability and sensor placement. *12th Intl. Work. on Principles of Diagnosis*.

Vignolles, A., Chanthery, E., & Ribot, P. (2020). An overview on diagnosability and prognosability for system monitoring. In *PHM Society European Conference* (Vol. 5).