Failure Mode and Effect Analysis for Abductive Diagnosis

Franz Wotawa[1]

Techn. Univ. Graz, Institute for Software Technology,
Inffeldgasse 16b/2, 8010 Graz, Austria
wotawa@ist.tugraz.at

Abstract. Diagnosis, i.e., fault localization in case of observing an unexpected behavior, is an important practical problem. During the past decades researchers have suggested several approaches for using models of the systems directly for identifying the root causes of failure. This model-based diagnosis approaches are either based on retaining consistency or on abduction. Despite their advantages both approaches are only rarely used in practical applications. In this paper, we focus on bringing abductive diagnosis closer to its application. In particular, we describe how failure mode and effect analysis, a technique of growing interest in applications, can be directly mapped to abductive diagnosis models. We discuss the basic foundations, and also problems that occur and how to deal with them. A direct conversion of FMEAs to abductive diagnosis problems would increase the use of abduction in practice because of avoiding writing logical theories directly.

Keywords: Abductive diagnosis, model-based diagnosis, FMEA

1 Introduction

Diagnosis as the task for identifying root causes explaining observations has been gaining lot of attention since the beginnings of Artificial Intelligence. The first diagnosis approaches like MYCIN [2] where based on expert systems, where knowledge has to be coded as rules that allow to derive causes from the observations. Later model-based approaches for diagnosis either using consistency-based approaches [7, 18] or abduction [4, 8, 5] has become very popular in research, which holds especially for consistency-based diagnosis. There are many advantages of model-based diagnosis. For example, the support of knowledge re-use or being able to derive root causes from the model itself, which is called reasoning from first principles. Despite the fact of all these advantages a broader application in practice is still missing. There are many reasons for this including the lack of available system models, the lack of modeling standards, and the intrinsic time and space complexity of model-based diagnosis.

More recently, industry uses more and more often failure mode and effect analysis (FMEA) [11, 17] as a framework for identifying critical faults that might occur. There the main focus is on the consequences of faults for the system's

behavior. FMEA are most often used for reliability analysis but also for other purposes like measuring diagnostic coverage [9]. One reason for the increasing use of FMEAs in practice is that there are standards like IEC61508, which require a detailed safety assessment (see for example [3]). Such an assessment for example is done on a regular basis within the automotive industry to ensure safety regulations. However, there are also other engineering areas where FMEAs are of growing importance, e.g., wind turbines [1].

Given the growing importance of FMEAs or similar analysis methods in industry and a growing demand for automated fault localization, the question is whether FMEAs can be directly used for the purpose of fault localization. It is worth mentioning that currently FMEAs are most often used for reliability analysis and not for fault localization. The main objective of this paper is to discuss the use of FMEAs for automated fault localization, which of course requires a FMEA to comprise essential information, i.e., at least the modes of replaceable parts responsible for effects and a description of effects, which can be observed. In particular, we show how adductive reasoning can be used for fault localization. Hence, in all cases where FMEAs are available that fit our purpose, we are able to use abductive diagnosis for practical applications, which widens the use of abductive reasoning in practice.

This paper is organized as follows. We first discuss the FMEA. For this purpose we introduce a small example that is going to be used in the rest of this paper. Afterwards, we briefly recall the basic definitions of abductive diagnosis. Finally, we show how FMEAs are mapped to abductive diagnosis problems, discuss some coding issues, and a property to hold in order to ensure effective diagnosis in practice. Moreover, in case of not fulfilling the property, we state several methods for solving the underlying problem. In the conclusion, we recapitulate the content of this paper and state our future research directions.

2 Failure mode and effect analysis

In the following we briefly introduce the FMEA but focus only on relevant parts we are going to use in this paper and thus ignoring all FMEA parts that belong to reliability analysis, e.g., assigning a likelihood to certain faults, the costs of the faults, and their severity. We focus on relevant parts only because the others do not provide any information used in the context of adductive diagnosis. We further assume that a FMEA describes the relationship between a fault and its effects ignoring intermediate effects or system internal non-observable effects. This is not a restriction because we are only interested in observable effects that might lead to dangerous situations. In cases where internal or intermediate effects are relevant they must have a corresponding observable effect, which has to considered in the FMEA.

We first, discuss FMEAs in more detailed using a simplified example coming from the wastewater treatment domain. In particular, we specify a FMEA for a constructed wetland. In Figure 1 we depict the structure of a constructed wetland. A constructed wetland is an artificial wetland used amongst others

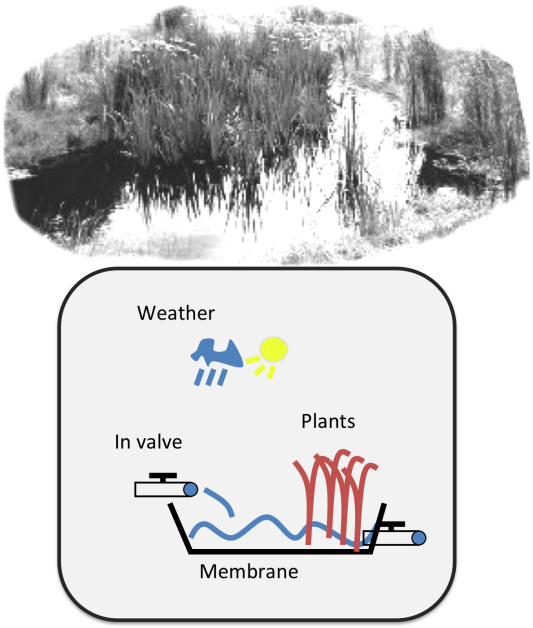

Fig. 1. A constructed wetland

for wastewater treatment. The idea behind the constructed wetlands is to use plants for the removal of nitrates and phosphate from polluted water. There are some parameters like the inflow of water, the status of the membrane used to prevent the water flowing out of the basin, and the outflow. All these parameters influence the water level. Moreover, also the weather conditions like rain, or a long period of heavy sunshine have an impact on the water level. On the other hand, the water level is important for ensuring the plant to survive. Is there is not enough water, the plant will finally die. Moreover, the concentration of nitrates and phosphates also impacts the health status of the plants. A healthy plant has a green color whereas a dying plant becomes brown. For more details about constructed wetlands as well as a discussion on the use of abductive reasoning versus decision trees we refer the interested reader to [20] and [19]. The latter discusses a different example from the wastewater treatment domain.

In order to construct a FMEA we first have to identify the components and potential fault modes. A constructed wetland comprises the components in-valve, out-valve, membrane, and plants. In addition we have to consider also external components, which also impacts the behavior. Such external components represent parts of the environment of a system that impacts the behavior. For the constructed wetland these components are the weather condition and the nitrates and phosphate (NP) concentration of the water that flows into the constructed wetland. The fault modes are wrong settings of the valve, a leak in

the membrane, a period of hot temperature and an unexpected NP concentration. It is worth noting that we are of course able to introduce more different fault modes like a lower inflow a higher inflow in order to distinguish different situations that are outside the operational settings of the constructed wetland. However, for simplicity of our example we only consider a few of them.

So, how can we construct a FMEA? A FMEA is a table, where each row describes the effects of a fault constituting because of a failure of a certain component. Hence, for each component and fault mode we make one entry. For example, from the verbal description of the constructed wetland, we see that a leaking membrane impacts the water level negatively, which in turn causes a plant to die because of lack of water. Hence, the color of the plants become brown. We are able to make a similar entry for each possible component and fault mode finally leading to the FMEA. For the constructed wetland the FMEA is given in Table 1. It is worth noting that in a FMEA usually only single faults are considered.

Table 1. A FMEA for a constructed wetland

Component	**Fault mode**	**Effect**
in-valve	too low	low inflow, wrong water level, brown plants
out-valve	too high	high outflow, wrong water level, brown plants
membrane	leaking	wrong water level, brown plants
weather	too hot	long sunshine period, wrong water level, brown plants
NP-conc.	wrong	brown plants

We now state a formal definition of a FMEA. We assume that a FMEA is for systems comprising components $COMP$, that have corresponding fault modes $MODES$. Moreover, we assume the existence of propositions $PROPS$ that belong to observations we obtained when running the system. If we observe a certain value, the proposition is said to be true, and false otherwise. We now define FMEAs as relation between components, a fault modes, and a set of effects that has to be observed when the given component is in the given fault mode. Note that in our constructed wetland example, there is a one to one relationship between components and modes. This is usually not the case in practice where a component like a logic gate has many fault modes. A fault model of a logic gate may at least comprise the fault modes stuck at one and stuck at zero.

Definition 1 (FMEA). *A FMEA is a set of tuples* (C, M, E) *where* $C \in COMP$ *is a component,* $M \in MODES$ *is a fault mode, and* $E \subseteq PROPS$ *a set of effects.*

For our constructed wetland example, we have to map the verbal descriptions given in Table 1 to elements of $COMP$, $MODES$, and $PROPS$. For this purpose we set $COMP = \{in, out, membrane, weather, np\}$, $MODES = \{low, high, leaking, hot, wrong\}$, and $PROPS = \{low\_in, high\_out, wrong\_level,$

$sunshine, brown\}$ and obtain the following FMEA for the constructed wetland example:

$$FMEA_{CW} = \left\{ \begin{array}{c} (in, low, \{low\_in, wrong\_level, brown\}) \\ (out, high, \{high\_out, wrong\_level, brown\}) \\ (membrane, leaking, \{wrong\_level, brown\}) \\ (weather, hot, \{sunshine, wrong\_level, brown\}) \\ (np, wrong, \{brown\}) \end{array} \right\}$$

In the next section, we briefly recapitulate the basic definitions of abductive diagnosis.

3 Abductive diagnosis

In this section we discuss the basic definitions of abductive reasoning and abductive diagnosis in particular. For this purpose, we first define a knowledge base comprising horn clause rules (from HC) over propositional variables $PROPS$. The restriction of logical formula to be horn is not a limitation in the context of physical systems since those models usually code the knowledge from causes to their effects. The used definitions are close the ones introduced by Friedrich at al. [8] and others in the area of abductive diagnosis [6, 5].

Definition 2 (Knowledge base (KB)). *A knowledge base (KB) is a tuple (A, Hyp, Th) where $A \subseteq PROPS$ denotes the set of propositional variables, $Hyp \subseteq A$ the set of hypothesis, and $Th \subseteq HC$ a set of horn clause sentences over A.*

In the definition of KB hypotheses corresponds directly to causes, such that for every cause there is a propositional variable that allows to hypothesize about the truth value of the cause. Hence, we use the terms hypotheses and causes in an interchangeable way.

Example 1. For example, the tuple $(\{wrong\_level, mode(in, low)\}, \{mode(in, low)\}, \{mode(in, low) \rightarrow wrong\_level\})$ forms a knowledge base. In the tuple the set $\{wrong\_level, mode(in, low)\}$ represents the used propositional variables, $\{mode(in, low)\}$ the hypotheses, and $\{mode(in, low) \rightarrow wrong\_level\}$ the set of clauses.

Definition 3 (PHCAP). *Given a knowledge base (A, Hyp, Th) and a set of observations $Obs \subseteq A$, the tuple (A, Hyp, Th, Obs) forms a propositional horn clause abduction problem (PHCAP).*

Example 2 (cont.). Assume that we observe a wrong water level, i.e., $Obs = \{wrong\_level\}$ we obtain a PHCAP.

Given a PHCAP we define a solution like in [8] as follows:

Definition 4 (Diagnosis; Solution of a PHCAP). *Given a PHCAP* (A, Hyp, Th, Obs)*. A set* $\Delta \subseteq Hyp$ *is a solution if and only if* $\Delta \cup Th \models Obs$ *and* $\Delta \cup Th \not\models \bot$*. A solution* Δ *is parsimonious or minimal if and only if no set* $\Delta' \subset \Delta$ *is a solution.*

A solution of a PHCAP is an explanation for the given observations. Instead of solution we also say abductive diagnosis (or diagnosis for short). In Definition 4 diagnoses need not to be minimal or parsimonious. In most practical cases only minimal diagnoses or minimal explanations for given effects are of interest. Hence, from here on we assume that all diagnoses are minimal diagnoses if not specified explicitly.

Example 3 (cont.). For our small example we obtain only one minimal diagnoses: $\Delta = \{mode(in, low)\}$ for the observation $wrong\_level$.

It is well known that the problem of finding minimal diagnoses for a given PHCAP is NP-complete. See for example Friedrich et al. [8] for a proof. Moreover, in the same paper the authors describe a general diagnosis and therapy process that make use of a PHCAP directly in order to identify and correct the detected faulty behavior. Unfortunately, we are not able to use Friedrich et al.'s algorithm because we require exactly one diagnosis before executing a repair action. Note that this holds in many cases where we want to diagnose engineered systems that behave outside the expectations. Of course, a single diagnosis might not always be required. Instead if the set of diagnosis is small a replacement of all potential diagnosis candidates can be performed. However, this can only be done if the set of diagnoses and their corresponding replacement costs are small enough. Therefore, we suggest to compute all possible diagnoses and to reduce this set by adding new observations that allow for discriminating diagnoses. This process is done until one diagnosis (or a diagnoses set that is small enough) is left and the corresponding treatment can be applied.

For adding new observations in order to decrease the number of diagnoses, we introduce the notation of discriminating observations.

Definition 5 (Discriminating observation). *Given a PHCAP* (A, Hyp, Th, Obs) *and two diagnoses* Δ_1 *and* Δ_2*. A new observation* $o \in A \backslash Obs$ *discriminates two diagnoses if and only if* Δ_1 *is a diagnosis for* $(A, Hyp, Th, Obs \cup \{o\})$ *but* Δ_2 *is not.*

In order to compute discriminating additional observations it is possible adopt the approach by De Kleer and Williams [14] who introduced an algorithm for measurement selection. For more details about the use of measurement selection for abductive diagnosis and an algorithm we refer to [19]. The underlying idea is to rank potential observations accordingly to their power of discrimination, e.g., preferring the observation that is able to divide the search space into two halves.

We implemented an algorithm for solving the PHCAP as well as the finding discriminating observations problem. This implementation is based on De Kleer's

Assumption-based Truth Maintenance System (ATMS) [12]. See for example [13] for an ATMS algorithm. An ATMS handles the consistency state of propositions that are connected via horn-clause rules. Moreover, an ATMS makes use of assumptions, which are assumed to be true unless a contradiction can be derived from the respective assumption. In ATMS terminology assumptions and propositions are said to be nodes, rules are justifications, and the contradiction ($\bot$) is represented by the special node NOGOOD. Besides the truth status an ATMS also manages for each node all the different assumptions that are necessary for making that node true. This set of assumptions is called label, and its elements an environment. An ATMS environment in the context of non-monotonic logic can be seen as a possible world that makes the corresponding node true.

An ATMS makes sure that for each node all the stored environments are consistent and each of them allow for deriving the node. Moreover, the environments are also minimal, and complete. The latter is important to ensure that there is not a possible world missing. An ATMS can be easily used for abductive diagnosis. Given a PHCAP. Let us first assume that all elements of A are mapped to nodes, all elements of Hyp are mapped to ATMS assumptions, and all rules in Th are mapped to ATMS justifications. If we want to obtain an observation $Obs = \{o\}$, then we only have to return the label l of the nodes that corresponds to o. All the elements of l are consistent and minimal. Moreover, from $l \cup Th$ we obtain o and there is no element missing in l. In general, the observations might contain more elements, i.e., $Obs = \{o_1, \ldots, o_k\}$. In this case we generate a new proposition σ and add $o_1 \wedge \ldots \wedge o_k \rightarrow \sigma$ to the theory Th that is passed to the ATMS. The label of the corresponding node of σ is an abductive diagnosis for Obs.

It is worth noting that the ATMS has been used in consistency-based diagnosis. The General Diagnostic Engine (GDE) [14] makes use of the ATMS for computing conflicts. A conflict in consistency-based diagnosis is a set of correctness assumptions for components that lead together with the logical description of a system and the given observations to a contradiction. When using an ATMS all possible conflicts are stored in the NOGOOD node. Reiter [18] proved that the hitting sets of minimal conflicts are the diagnoses. In his paper, Reiter also provided an algorithm for computing the diagnoses from the conflicts. Greiner et al. [10] provided an improved version of Reiter's hitting set algorithm. There have been many applications of the GDE or its variant for diagnosis including [16, 15]. In contrast to adductive diagnosis used in this paper, the original consistency-based diagnosis approach considers the correct behavior of a system but not models of faulty behavior. Moreover, the GDE makes us of the information stored in the NOGOOD node, whereas we are interested in the assumptions that are stored in the node, which corresponds to σ.

In the next section, we discuss how to make use of a FMEA for obtaining a corresponding KB further used to construct a PHCAP. We also discuss some coding issues because of the restriction to horn-clause propositional logic.

4 Mapping FMEAs to abductive diagnosis

In this section we discuss the mapping of FMEA to a corresponding (abductive) KB, which can be used for diagnosis purposes. Afterwards we discuss some properties of the knowledge base, which must hold in order to ensure the computability of a single diagnosis in case enough information (i.e., observations) are given. Furthermore, we show how FMEA can be revised in order to guarantee the computation of single diagnosis.

Let us first start with the mapping of FMEAs to their corresponding knowledge base KB. This mapping is rather straightforward because FMEA already capture the cause-effect relationship in tabular form. Formally, we use a function $\mathfrak{M} : 2^{FMEA} \mapsto HC$ for mapping a particular FMEA to a horn clause theory. In the following we define $\mathfrak{M}$ in the following way iterating over all tuples stored in FMEA:

Definition 6 (FMEA to HC; $\mathfrak{M}$). *Given a FMEA, the function $\mathfrak{M}$ is defined as follows:*

$$\mathfrak{M}(FMEA) =_{def} \bigcup_{t \in FMEA} \mathfrak{M}(t)$$

where

$$\mathfrak{M}(C, M, E) =_{def} \{mode(C, M) \rightarrow x \,|x \in E\}.$$

The causes, i.e., the components and their modes, stored in an FMEA, are mapped to proposition. Hence, we define the set of hypotheses as follows:

$$Hyp =_{def} \bigcup_{(C,M,E) \in FMEA} \{mode(C, M)\}.$$

What remains is the set of proposition. This set is equivalent to the union of all effects stored in a FMEA, i.e.:

$$A =_{def} \bigcup_{(C,M,E) \in FMEA} E.$$

Example 4. When using the $FMEA_{CW}$ of the constructed wetland given at the end of Section 2 we obtain the following corresponding KB_{CW} when applying the $\mathfrak{M}$ function and the other definitions for Hyp and A:

$$Hyp = \left\{ \begin{array}{c} mode(in, low), mode(out, high), mode(membrane, leaking), \\ mode(weather, hot), mode(np, wrong) \end{array} \right\}$$

$$A = \{low\_in, high\_out, sunshine, wrong\_level, brown, \}$$

$$Th = \left\{ \begin{array}{c} mode(in, low) \rightarrow low\_in, mode(in, low) \rightarrow wrong\_level, \\ mode(in, low) \rightarrow brown, mode(out, high) \rightarrow high\_out, \\ mode(out, high) \rightarrow wrong\_level, mode(out, high) \rightarrow brown, \\ mode(membrane, leaking) \rightarrow wrong\_level, \\ mode(membrane, leaking) \rightarrow brown, mode(weather, hot) \rightarrow sunshine, \\ mode(weather, hot) \rightarrow wrong_level, mode(weather, hot) \rightarrow brown, \\ mode(np, wrong) \rightarrow brown \end{array} \right\}$$

When using KB together with the observation that the plants are brown, i.e., $Obs = \{brown\}$, all the elements of Hyp are also a (single) diagnosis. Hence, someone is interested in reducing the number of diagnoses by adding new observations.

A reduction of diagnosis candidates can be done – as already discussed – using additional observations. For example, when adding the observation low_in to Obs in Example 4, only the diagnosis $\{mode(in, low)\}$ remains. In practice the observation low_in would be obtained by inspecting the in-valve and comparing its flow with the expected flow. But what happens if the inflow is correct? In this case we have to add $\neg low\_in$ to Obs and $\{mode(in, low)\}$ would be removed from the list of diagnosis candidates because $\{mode(in, low)\} \cup Th \not\models \{brown, \neg low\_in\}$. However, also the other diagnosis candidates would be removed, because non of them lead to the derivation of $\neg low\_in$.

One way of overcoming this problem is to model also negated observations. However, a straightforward implementation would not be possible if we still rely on horn-clause logic. Hence, we would have to introduce special propositions, e.g., not_low_in, representing the fact $\neg low\_in$. In addition we would have to add inconsistency checking rules, stating that a proposition and its negation cannot be true at the same time, e.g., $low\_in \wedge not\_low\_in \rightarrow \bot$. Such a solution increases the size of the KB.

Another solution would be to state that a certain property does not longer hold. For example, if low_in has not been observed, we are able to simple state that this property in the current world would lead to an inconsistency, e.g., $low\_in \rightarrow \bot$. In this way, we rule out all diagnoses that derives an observation we are not able to establish in reality. In our system we follow this idea and represent negated observations o as additional rules of the form $o \rightarrow \bot$ in the KB's theory Th.

What is missing is to give an answer to the question whether the used model and in this case the underlying FMEA is good enough for diagnostic purposes. By 'good enough' we mean that the obtained results are correct and complete. This is ensured using abductive diagnosis where only correct, i.e., consistent, diagnoses are computed. Moreover, when searching for all diagnoses, we are sure that completeness is obtained with respect to the underlying model. Hence, we define 'good enough' for diagnosis here as the ability of the KB to find one single fault diagnosis if all necessary information is given. We call this property one single fault diagnosis property and define it as follows:

Definition 7 (One Single Fault Diagnosis Property (OSFDP)). *Given a KB (A, Hyp, Th). KB fulfills the One Single Diagnosis Property (OSDP) if the*

following hold:

$\forall m \in Hyp : \exists Obs \subseteq A : \{m\}$ *is a diagnosis of* (A, Hyp, Th, Obs) *and*
$\neg\exists m' \in Hyp : m' \neq m$ *such that* $\{m'\}$ *is a diagnosis for the same PHCAP.*

The OSFDP property allows us to ensure that with enough knowledge we are able to distinguish all single-fault diagnosis. If a KB does not fulfill OSFDP, then we cannot distinguish all diagnoses. Hence, in some cases we need to replace more components. Checking for OSFDP can be done in a simple but effective way. We only have to go through all hypotheses and compute all proposition that follows from the current hypothesis and the theory Th. Afterwards we have to check for two hypotheses whether their derived propositions are the same. If yes, then the OSFDP can never be fulfilled. If we do not find such a pair, then the OSFDP has to hold because we can distinguish the two hypotheses with the non-intersecting parts of their propositions. In Algorithm 1 we summarize the checking procedure.

Algorithm 1 CHECK_OSFDP(A, Hyp, Th)

Input: A KB (A, Hyp, Th) comprising propositions A, hypothesis Hyp, and a horn-clause propositional logic.
Output: True if the KB fulfills the OSFDP, and false otherwise.

```
1: for all h ∈ Hyp do
2:    Let O ⊆ A be the largest set such that {h} ∪ Th ⊨ O, if {h} ∪ Th ⊨ ⊥, and ∅
      otherwise.
3:    Let δ(h) be O.
4: end for
5: for all h1 ∈ Hyp and δ(h1) ≠ ∅ do
6:    for all h2 ∈ Hyp and h1 ≠ h2 ∧ δ(h2) ≠ ∅ do
7:       if δ(h1) = δ(h2) then
8:          return false
9:       end if
10:   end for
11: end for
12: return true
```

Algorithm 1 obviously terminates assuming Hyp to be finite (which is always the case in practice). The algorithm's time complexity is polynomial and in particular of order $O(|Hyp|^2)$. The runtime is mainly determined by the nested loops (line 5 to 10). Note also that the algorithm requires too many checks, which might be avoided. However, the time complexity would still remain the same considering the $O(.)$ notation.

Example 5. The KB from Example 4 fulfills the OSFDP. Algorithm 1 would return $true$.

It is worth noting that – because of the construction of $\mathfrak{M}$ – the check for OSFDP can also be easily checked on side of the FMEA. For this purpose only the effects of the given causes have to be compared with each other. We will use this observation for the further elaboration on the question of how to deal with KBs not fulfilling OSFDP. In principle, we have three possibilities:

1. We ignore that OSFDP is not fulfilled. In this case we know that there are diagnoses, which we cannot distinguish regardless of the given observations.
2. If we know that there are two hypotheses h_1 and h_2 contradicting the OSFDP, then there is no way for distinguishing h_1 from h_2. Hence, we can treat both hypotheses the same. For this purpose, we introduce a new hypothesis h' and replace both h_1 and h_2 with h. We proceed with this replacements until the OSFDP is fulfilled. This method summarizes all hypotheses that cannot be distinguished and which should be treated as a single replaceable unit.
3. It is worth noting that there is another reason why an OSFDP is not fulfilled. There might be other observations of the system or internal values not observable from outside that are currently not considered in the given FMEA. By introducing these observables into the FMEA, we might be able to distinguish hypotheses. Hence, the third way of dealing with the problem is to ask the user for further observations to be used in the FMEA.

Hence, with the described methods the OSFDP can either be handled as it is, or treated in a way that allows for coming up with a FMEA (or the corresponding KB) where the OSFDP is fulfilled.

5 Conclusions

In the paper we argued that abductive reasoning and in particular abductive diagnosis become more important for practical diagnostic applications. This is due to the fact of the growing importance of FMEAs used in industry for ensuring and also evaluating reliability and safety constraints. We discussed a method for mapping FMEAs directly to knowledge bases that code the knowledge in rules representing the cause-effect relationship directly. Furthermore, we also discussed how this knowledge base can be used for diagnosis and also give some information on how to deal with observations. In particular, we discussed the problem of handling negated observations without changing the knowledge base substantially. We do this by adding a rule stating that the observation would lead to an inconsistency.

Finally, we introduced a property that has to hold for knowledge bases in order to allow for computing exactly one single-fault diagnosis if there is enough information provided. We also discussed what to do in case the knowledge base does not fulfill the property. There are two active ways of dealing with the problem: (1) handling non-distinguishable single-fault diagnosis as one component, and (2) asking the user for more observation and extensions for the FMEA.

The former method generates a super-component that can be seen as a replaceable unit in case of failing. The latter might be interpreted in the way that the modeling has to be improved.

Future research will include an implementation of the proposed techniques and algorithms. Currently, the abductive diagnosis is implemented based on an ATMS implementation. Moreover, we want to use this method for diagnosis of technical systems. The first application will be in the context of wind-turbines used to generate electricity, where the FMEA comes from our industrial partner. Using the application domain we want to clarify whether abductive reasoning is a good foundation for diagnosis or not. This includes computational aspects as well as usability concerns. With the automated conversion of FMEAs to KBs, we believe to have taken, an important step towards increased usability.

Acknowledgements. The research presented in this paper has received funding from the Austrian Research Promotion Agency (FFG) under grant 842407 (Applied MOdel-based Reasoning (AMOR)). Many thanks also go to the anonymous reviewers for their very valuable comments helping to improve this paper.

References

1. Arabian-Hoseynabadi, H., Oraee, H., Tavner, P.: Failure modes and effects analysis (fmea) for wind turbines. Electrical Power and Energy Systems 32, 817–824 (2010)
2. Buchanan, B.G., Shortliffe, E.H. (eds.): Rule-Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley Publishing Company (1984)
3. Catelani, M., Ciani, L., Luongo, V.: The fmeda approach to improve the safety assessment according to the iec61508. Microelectronics Reliability 50, 1230–1235 (2010)
4. Console, L., Dupré, D.T., Torasso, P.: A theory of diagnosis for incomplete causal models. In: Proc. IJCAI. pp. 1311–1317. Morgan Kaufmann, Detroit (Aug 1989)
5. Console, L., Dupré, D.T., Torasso, P.: On the relationship between abduction and deduction. Journal of Logic and Computation 1(5), 661–690 (1991)
6. Console, L., Torasso, P.: Integrating models of correct behavior into abductive diagnosis. In: Proceedings of the European Conference on Artificial Intelligence (ECAI). pp. 160–166. Pitman Publishing, Stockholm (Aug 1990)
7. Davis, R., Shrobe, H., Hamscher, W., Wieckert, K., Shirley, M., Polit, S.: Diagnosis based on structure and function. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). pp. 137–142. Pittsburgh (Aug 1982)
8. Friedrich, G., Gottlob, G., Nejdl, W.: Hypothesis classification, abductive diagnosis and therapy. In: Proceedings of the International Workshop on Expert Systems in Engineering. Springer Verlag, Lecture Notes in Artificial Intelligence, Vo. 462, Vienna (Sep 1990)
9. Goble, W., Brombacher, A.: Using a failure modes, effects and diagnostic analysis (fmeda) to measure diagnostic coverage in programmable electronic systems. Reliability Engineering & System Safety 66, 145–148 (1999)
10. Greiner, R., Smith, B.A., Wilkerson, R.W.: A correction to the algorithm in Reiter's theory of diagnosis. Artificial Intelligence 41(1), 79–88 (1989)

11. Hawkins, P.G., Woollons, D.J.: Failure modes and effects analysis of complex engineering systems using functional models. Artificial Intelligence in Engineering 12, 375–397 (1998)
12. de Kleer, J.: An assumption-based TMS. Artificial Intelligence 28, 127–162 (1986)
13. de Kleer, J.: A general labeling algorithm for assumption-based truth maintenance. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). pp. 188–192. Morgan Kaufmann, Saint Paul, Minnesota (Aug 1988)
14. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. Artificial Intelligence 32(1), 97–130 (1987)
15. Malik, A., Struss, P., Sachenbacher, M.: Qualitative modeling is the key – a successful feasibility study in automated generation of diagnosis guidelines and failuer mode and effects analysis for mechatronic car subsystems. In: Proceedings of the Sixth International Workshop on Principles of Diagnosis (1995)
16. Malik, A., Struss, P., Sachenbacher, M.: Case studies in model-based diagnosis and fault analysis of car-subsystems. In: Proceedings of the European Conference on Artificial Intelligence (ECAI) (1996)
17. Price, C., Taylor, N.: Automated multiple failure fmea. Reliability Engineering & System Safety 76, 1–10 (2002)
18. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)
19. Wotawa, F., Rodriguez-Roda, I., Comas, J.: Environmental decision support systems based on models and model-based reasoning. Environmental Engineering and Management Journal 9(2), 189–195 (2010)
20. Wotawa, F.: On the use of abduction as an alternative to decision trees in environmental decision support systems. International journal of agricultural and environmental information systems 2(1), 63–82 (2011)