In *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW´96)*, Banff,

Canada, November 9th - November 14th, 1996.

Assumptions in Model-based Diagnosis

Dieter Fensel and Richard Benjamins

University of Amsterdam, Department SWI, Roetersstraat 15, 1018 WB Amsterdam, the Netherlands,
E-mail: {fensel | richard}@swi.psy.uva.nl

Abstract. Mostly, papers on problem-solving methods focus on the description of reasoning strategies and discuss their underlying assumptions as a side aspect. We take a complementary point of view and focus on these underlying assumptions as they play three important roles: first, assumptions are necessary to characterise the precise competence of a problem-solving method in terms of the tasks that can be solved by it, and in terms of the domain knowledge that is required by it. Second, assumptions are necessary to enable tractable problem solving for complex problems. Third, assumptions are necessary for appropriate interaction of the problem solver with its environment. Their introduction and refinement can be used to develop new problem-solving methods, or to adapt existing ones according to task and domain-specific circumstances of a given application. For this purpose, one requires a framework for dealing with these assumptions. This paper makes a step in this direction by summarising the assumptions that can be found in the literature on diagnosis with component-oriented device models. The main contribution of the paper is to collect these assumptions, to make their role in the reasoning process explicit, and to classify them.

1 There Is No Such Thing As An Assumption-Free Reasoning Strategy

Reasoning about real-world problems is only possible by introducing assumptions about these problems. Such assumptions are necessary to relate the input and the output of the reasoning process with the actual problem. Assumptions refer to the provided case data and the precise task that should be executed by the reasoner. In the case of knowledge-based reasoners, additional assumptions appear that are formulated as requirements on knowledge as it is necessary for the reasoning process. Another type of assumptions deals with the complexity of the reasoning task. In general, most problems tackled with knowledge-based systems are inherently complex and intractable ([Bylander, 1991], [Bylander et al., 1991], [Nebel, 1996]). A knowledge-based reasoner can only solve such tasks with reasonable computational effort by introducing assumptions that restrict the complexity of the problem, or strengthen the requirements on domain knowledge (cf. [Benjamins et al., 1996]). Therefore, this second type of assumptions either weakens the task definition or introduces additional requirements on the domain knowledge that is expected by the problem solver. In the first case, the task is weakened (i.e., the application scope is reduced) to meet the competence of the reasoning system and in the second case, the competence of the reasoner is strengthened by the assumed domain knowledge.

In [Fensel & Straatman, 1996] we discussed the idea of viewing the development process of an efficient knowledge-based reasoner as a process of introducing and refining assumptions. A simple generate & test problem solver is modified to a more efficient problem-solving method (a variant of propose & revise for parametric design[1] by introducing assumptions that weaken the task and that strengthen the requirements on domain knowledge. In this paper, we review the work on diagnostic reasoning systems. We focus on assumptions that are introduced to define the effect of diagnostic reasoners and to improve their efficiency. The work on diagnostic reasoning provides an interesting empirical basis for our approach as it provides a more than ten years long history on developing knowledge-based reasoners for a complex task. During this process, several assumptions have been identified and refined in the literature on model-based diagnostic systems.

The first diagnostic systems built were heuristic systems, in the sense that they contained compiled knowledge which linked symptoms directly to hypotheses (usually in rules). In these systems, only foreseen symptoms can be diagnosed and heuristic knowledge that links symptoms with possible faults needs to be available. One of the main principles underlying model-based diagnosis is the use of a domain model (called SBF models in [Chandrasekaran, 1991]). Heuristic knowledge that links symptoms with causes is no longer necessarily. The domain model is used for predicting the desired device behaviour, which is then compared to the observed device behaviour. A discrepancy indicates a symptom. General reasoning techniques as constraint satisfaction or truth maintenance can be used to derive diagnoses that explain the actual behaviour of the device using its model. Because the reasoning part is represented separately from domain knowledge, it can be reused for different domains. This paradigm of model-based diagnosis gave rise to the development of general approaches to diagnosis, such as "constraint suspension" [Davis, 1984], DART [Genesereth, 1984], GDE [de Kleer & Williams, 1987], and several extensions to GDE (GDE+ [Struss & Dressler, 1989], Sherlock [de Kleer & Williams, 1989]).

In this paper, we will focus on assumptions underlying approaches to diagnostic problem solving. In Section 2, we discuss assumptions that are necessary to relate a diagnostic system with its environment. That is, assumptions on the available case data, the required domain knowledge and the problem type. In Section 3, we discuss assumptions introduced to reduce the complexity of the reasoning process necessary to execute the diagnostic task. Such assumptions are introduced to either change the worst-case complexity or the average-case behaviour of problem solving. In Section 4, we sketch further assumptions that are related to the efficiency of the interaction of the problem solver with its environment. Section 5 sketches a general framework for specifying reasoning systems and their underlying assumptions. It discusses how assumptions can be used to weaken or strengthen the problem-solving method competence. In section 6, we present conclusions.

2 The Task: Assumptions for Effect

In model-based diagnosis (cf. [de Kleer et al., 1992]), the definition of the task of the KBS requires a *system description* of the device under consideration and a *set of observations*, where some indicate *normal* and other *abnormal* behaviour. The goal of the task is to find a *diagnosis* that, together with the system description, *explains* the observations. In the following, we discuss four different aspects of such a task definition and show the assumptions related to each of them. The four aspects are: identifying abnormalities, identifying causes of these abnormalities, defining hypotheses, and defining diagnoses.

2.1 Identifying Abnormalities

Identification of abnormal behaviour is necessary before a diagnostic process can be started to find explanations for the abnormalities. This identification task requires three kinds of knowledge, of which two are related to the type of input, and one to the interpretation of possible discrepancies (see [Benjamins, 1993]):

---

1. This problem-solving method propose & revise was analysed versus its underlying assumptions and their rationale in [Fensel, 1995].

- *observations* of the behaviour of the device must be provided to the diagnostic reasoner;
- a *behavioural description* of the device must be provided to the diagnostic reasoner;
- knowledge concerning the (*im*)*preciseness* of the observations and the behavioural description as well as *comparison knowledge* (thresholds etc.) are necessary (for comparison) to decide whether a discrepancy is significant. Other required knowledge concerns the interpretation of *missing values*, and whether an observation can have several values (i.e., its value type).

Relevant assumptions state that the two types of inputs need to be *reliable*. Otherwise, the discrepancy could be explained by a measuring fault or a modelling fault. In other words, these assumptions guarantee that if a prediction yields a different behaviour than the observed behaviour of the artefact, then the artefact has a defect [Davis & Hamscher, 1988]).

These assumptions are also necessary for the meta-level decision whether a diagnosis problem is given at all (i.e., whether there is an abnormality in system behaviour). This decision relies on a further assumption: the *no-design-error assumption* [Davis, 1984] which says that if no fault occurs, then the device must be able to achieve the desired behaviour. In other words, the discrepancy must be the result of a fault situation where some parts of the system are defect. It cannot be the result of a situation where the system works correctly, but cannot achieve the desired functionality because it is not designed for this. If this assumption does not hold, one has a design problem and not a diagnostic problem.

2.2 Identifying Causes

Another purpose of the system description is the identification of causes of faulty behaviour. This cause-identification knowledge must be *reliable* [Davis & Hamscher, 1988], or, in other words, the knowledge used in model-based diagnosis is assumed to be a correct and complete description of the artefact. Correct and complete in the sense, that it enables the derivation of correct and complete diagnoses if discrepancies appear.[2] In accordance with different types of device models and diagnostic methods, these assumptions wear different clothes. During the following we restrict our attention to component-oriented device models that describe a device in terms of components, their behaviours (a functional description), and their connections.[3] The set of all possible hypotheses is the power-set of the set of annotated components

$$\{mode_{11}(c_1), mode_{12}(c_1), ..., mode_{nm_n}(c_n)\},$$

where $mode_{ji}(c_j)$ describes that the j-th component is in mode i. [Davis, 1984] has pointed out that one should be aware of the underlying assumptions for such a diagnostic approach and listed a number of them.

First, the *localised-failure-of-function* assumption: the device must be decomposable in well-defined and localised entities (i.e., a component) that can be treated as causes of faulty behaviour. Second, these *components have a functional description* that provides the (correct) output for their possible inputs. If this functional description is local, that is, it does not refer to the functioning of the whole device, the *no-function-in-structure* assumption [de Kleer & Brown, 1984] is satisfied. Several diagnostic methods can also use the reverse of the functional descriptions, thus, rules that *derive the expected input from the provided output*. If only correct functional descriptions are available, fault behaviour is defined as any other behaviour than the correct one. Fault behaviour of components can be constrained by including fault models, that is, *functional descriptions of the components in case they are broken* (cf. [de Kleer &

2. A typical problem of diagnosis without knowledge about fault models (i.e., incomplete knowledge) is that the reasoner provides in addition to the right diagnoses also wrong diagnoses. The derived result is complete but not correct because the provided domain knowledge is not complete. No correct domain knowledge can lead to incomplete sets of diagnoses.

3. It is a critical modelling decision what to view as a component and which types of interactions are represented (cf. [Davis, 1984]). Several points of view are possible to decide what is regarded as being a component. Different levels of physical representations result in different entities; the independent entities that are used in the manufacturing process of the artefact could be used as components; or functional unities of the artefact could be seen as components.

Williams, 1989], [Struss & Dressler, 1989]). If one assumes that these functional descriptions are complete (the *complete fault-knowledge* assumption), then components can be considered innocent if none of their fault descriptions is consistent with the observed fault behaviour. A result of using fault models is that all kinds of non-specified behaviours of a component are excluded as diagnosis. For example, using fault models, it becomes impossible to conclude that a fault (one of two light bulbs is not working) is explained by a defect battery that does not provide power and a defect lamp that lights without electricity (cf. [Struss & Dressler, 1989]).

Further assumptions that are related to the functional descriptions of components are the *no-fault-masking* and the *non-intermittency* assumption. The former states that the defect of an individual or composite component, or of the entire device must be visible by changed outputs (cf. [Davis & Hamscher, 1988], [Raiman, 1992]). According to the latter, a component that gets identical inputs at different points of time, must produce identical outputs. In other words, the output is a function of the input (cf. [Raiman et al., 1991]). They argue that intermittency results from incomplete input specifications of components, but that it is impossible to get rid of it (it is impossible to represent the required additional inputs).

A third assumption underlying many diagnostic approaches is the *no-faults-in-structure* assumption (cf. [Davis & Hamscher, 1988]) that manifests itself in different variants according to the particular domain. The assumption states that the interactions of the components are correctly modelled and that they are complete. This assumption gives rise to three different classes of more specific assumptions. First, the *no-broken-interaction* assumption states that connections between the components work correctly (e.g. no wires between components are broken).[4] If this is not the case, the assumption can be weakened by representing the connections themselves as components too. Second, the *no-unexpected-directions* assumption (or existence of a causal-pathway assumption) states that the directions of the interactions are correctly modelled and are complete. For example, a light-bulb gets power from a battery and there is no interaction in the opposite direction. The *no-hidden-interactions* assumption (cf. [Böttcher, 1996]) assumes that there are no non-represented interactions (i.e., closed-world assumptions on connections). A bridge fault [Davis, 1984] is an example of a violation of this assumption in the electronic domain. Electronic devices whose components unintendedly interact through heat exchange, is another example [Böttcher, 1996]. In the worst case, all potential unintended interaction paths between components are represented [Preist & Welhalm, 1990]. The no-hidden-interactions assumption is critical since most models (like design models of the device) describe correctly working devices and unexpected interactions are therefore precisely not mentioned. A refinement of this assumptions is that there are no *assembly errors* (i.e., every individual component works but they have been wired up incorrectly).

2.3 Defining Hypotheses

In addition to knowledge that is required to identify a discrepancy and knowledge that provides hypotheses used to explain these discrepancies, one requires further knowledge to decide which type of explanation is required. [Console & Torasso, 1992] distinguish two types of explanations: weak explanations, that are *consistent* with the observations (no contradiction can be derived from the union of the device model, the observations, and the hypothesis), and strong explanations, that *imply* the observations (the observations can be derived from the device model and the hypothesis). Both types of explanation can be combined by dividing observations in two classes: observations that need to be explained by deriving them from a hypothesis, and observations that need only be consistent with the hypothesis. In this case one requires *knowledge that allows to divide the set of observations*. The decision which type of explanation to use, can only be made based on assumptions about the environment in which the KBS is used.

---

4. It is possible to represent the interactions between components as possible hypotheses but this leads to new problems (compare Section 2.5).

2.4 Defining Diagnoses

Having established observations, hypotheses and an explanatory relation that relates hypotheses with observations, one must establish the notion of *diagnosis*. Not each hypothesis that correctly explains all observations needs to be a desired diagnosis. One could accept only *parsimonious* hypotheses as *diagnoses* (cf. [Bylander et al., 1991]). An hypothesis or explanation H is parsimonious if H is an explanation and there exists no other hypothesis H' that also is an explanation and $H' < H$. H is a diagnosis if H is a parsimonious explanation. One has to make assumptions about the desired diagnosis (cf. [McIlraith, 1994]) in order to define the partial ordering (<) on hypotheses. For example, whether the diagnostic task is concerned with finding all components that are necessarily fault to explain the system behaviour, or whether it is concerned with finding all components that are necessarily correct to explain the system behaviour. In the first case, we aim at economy in repair, whereas in safety critical applications (e.g., monitoring of nuclear power plants) one should obviously choose for the second case.

As shown by [McIlraith, 1994], the assumptions about the type of explanation relation (i.e., consistency versus derivability) and about the explanations (i.e., definition of parsimony) make also strong commitments on the domain knowledge (the device model) that is used to describe the system. If we ask for a consistent explanation with minimal sets of fault components (i.e., $H_1 < H_2$ if H_1 assumes less components as being fault than H_2), we need knowledge that constrains the normal behaviour of components. If we ask for a consistent explanation with minimal sets of correct components (i.e., $H_1 < H_2$ if H_1 assumes less components as being correct than H_2), we need knowledge that constrains the abnormal behaviour of components.

The definition of parsimonious hypotheses introduces a *preference* on hypotheses. This could be extended by defining further preferences on diagnoses to select one optimal one (e.g., by introducing assumptions related to the probability of faults). Again, knowledge about preferences must be available to define a preference function and a corresponding ordering.

2.5 Summary

Figure 1 summarises the assumptions that are discussed above and groups them according to their purpose. All these assumptions are necessary to relate the definition of the functionality of the diagnostic system with the diagnostic problem (i.e., the task) to be solved and with the domain knowledge that is required to define the task. Table 1 provides an explanation of the assumptions along with the role they play (function), the domain they are about (case data, domain knowledge or task), and some references where they are discussed in more detail.

Table 1: Effect-Assumptions in component-oriented diagnosis.

name	explanation	is about[a]	function	some references
existence of observations	observations must be provided to the system	cd	It is necessary for detecting discrepancies.	[Benjamins, 1993]
reliability of observations	The provided observations must be reliable.	cd	It is necessary for assuming that the discrepancy must be explained by a diagnosis.	[Benjamins, 1993], [Davis & Hamscher, 1988]
existence of a behavioural description	The desired system behaviour must be known to the diagnostic reasoner.	dk	It is necessary for detecting discrepancies.	[Benjamins, 1993]
reliability of behavioural description	The description of the system must be reliable.	dk	It is necessary for assuming the discrepancy must be explained by a diagnosis.	[Benjamins, 1993], [Davis & Hamscher, 1988]

Table 1: Effect-Assumptions in component-oriented diagnosis.

name	explanation	is about[a]	function	some references
existence of fault identification knowledge	Knowledge is required to compare the observations with the behavioural description.	dk	It is necessary for interpreting discrepancies.	[Benjamins, 1993]
reliability of the fault identification knowledge	The knowledge used to detect discrepancies must be reliable.	dk	It is necessary for interpreting discrepancies correctly.	[Benjamins, 1993]
no-design-error	The discrepancy between expected and actual behaviour does not result from the design of the device.	t	The behavioural discrepancy is a fault and not just an impossibility.	[Davis, 1984]
existence of a set of components	The device can be decomposed into a set of components.	dk	The entire device can be decomposed into smaller units that constitute the device.	[Davis, 1984], [Davis & Hamscher, 1988]
localized-failure-of-function, no-function-in-structure	Fault components can be identified as causes.	dk	The reasons for fault behaviour do not have to be constructed but can be selected from a finite set.	[Davis, 1984], [de Kleer & Brown, 1984]
existence of a set of annotations	Components could have several behavioural modes that need to be provided.	dk	The diagnostic reasoner can select from the behavioural modes provided for each component.	[Struss & Dressler, 1989], [de Kleer et al., 1992]
completeness of the set of annotations = complete-fault knowledge	All possible modes of the components are known.	dk	It is used to infer the mode of a component if all other behaviours do not (even not partially) explain the fault.	[Struss & Dressler, 1989], [de Kleer et al., 1992]
existence of input-output descriptions of the components	This knowledge defines the input-output behaviour of the components.	dk	The behavioural description of the components is required to detect their faulty behaviour and to derive the overall behaviour of the complete device.	[de Kleer & Williams, 1987], [Davis & Hamscher, 1988]
existence of output-input descriptions of the components	This knowledge defines the output-input relation of the components.	dk	This knowledge can be used to derive additional discrepancies.	[Davis, 1984], [Raiman, 1989]
existence of functional descriptions of faulty behaviour of components	This knowledge defines the input-output behaviour of the components in the case a fault occurs.	dk	The behavioural description of the components is required to identify different possible faults of a component.	[de Kleer & Williams, 1987], [Struss & Dressler, 1989]
complete behavioural descriptions (complete fault-models)	All possible behaviours of a component is modelled by its functional description.	dk	It is used to completely constrain the possible behaviour of a component.	[de Kleer & Williams, 1987], [Struss & Dressler, 1989]
no-fault-masking	A fault of a component is visible in its behaviour and in the behaviour of the entire device.	cd & dk	It is necessary for detecting faulty components.	[Davis, 1984], [Davis & Hamscher, 1988], [Raiman, 1992]
non-intermittency	The output of a component is a function of the input (e.g., the behaviour does not change over time).	cd	It is necessary for interpreting the discrepancy between an observation and an output of a behavioural description of a component.	[Davis, 1984], [Raiman et al., 1991]

Table 1: Effect-Assumptions in component-oriented diagnosis.

name	explanation	is about[a]	function	some references
existence of a model of the component interactions	It assumes that the possible interactions between components are known to the reasoner.	dk	This model is required to derive the overall behaviour of the system and the local inputs of components from the local outputs of the components.	[Davis, 1984], [Davis & Hamscher, 1988]
no-fault in structure assumption	Faulty components are the only causes.	dk	Only components need to be treated as possible causes for the faulty behaviour.	[Davis, 1984], [Davis & Hamscher, 1988]
no-broken-interactions	The interactions work properly, i.e., the connections work properly.	dk	Only components need to be treated as possible causes for the faulty behaviour and the interaction model describes the real interactions.	[Davis, 1984], [Davis & Hamscher, 1988]
no-unexpected direction	The direction of the interaction is as represented.	dk	Only components need to be treated as possible causes for the faulty behaviour and the interaction model describes the real interactions.	[Davis, 1984]
no-hidden interactions (closed world assumptions)	There are no interactions that are not represented in the model.	dk	Only components need to be treated as possible causes for the faulty behaviour and the interaction model describes the real interactions.	[Davis, 1984], [Böttcher, 1996]
no assembly error	The components are not wired incorrectly.	dk	Only components need to be treated as possible causes for the faulty behaviour and the interaction model describes the real interactions.	[Davis & Hamscher, 1988], [Böttcher, 1996]
type of explanation relation (type of hypotheses)	Need an observation be consistent with the diagnosis or must it be derivable from it.	dk & t	The problem solving is either constraints satisfaction or abductive inference.	[Console & Torasso, 1992], [de Kleer et al., 1992], [ten Teije & van Harmelen, 1996]
classification of observations	It introduces an distinction between observation describing normal and abnormal behaviour.	dk	In abductive inference only the abnormal behaviour must be explained.	[Console & Torasso, 1992]
type of explanation (type of diagnosis)	Should the set of fault components contain all components that need to be fault or that could be fault.	dk & t	Economy in repair versus safety-critical monitoring.	[McIlraith, 1994]
preference knowledge on diagnoses	It defines preferences between diagnoses.	dk	Necessary for selecting the diagnoses with high preferences.	[de Kleer & Williams, 1987], [Davis & Hamscher, 1988]

a. cd = case data, dk = domain knowledge, t = task

All these assumptions are necessary to relate a model of the device with the actual device under concern. “There is no such thing as an assumption-free representation. Every model, every representation contains simplifying assumptions“ [Davis & Hamscher, 1988]. If the assumptions are too strong, one could consider weakening them.[5] However, this raises another problem in model-based diagnosis, namely its *high complexity or intractability*. This will be discussed in the following section.

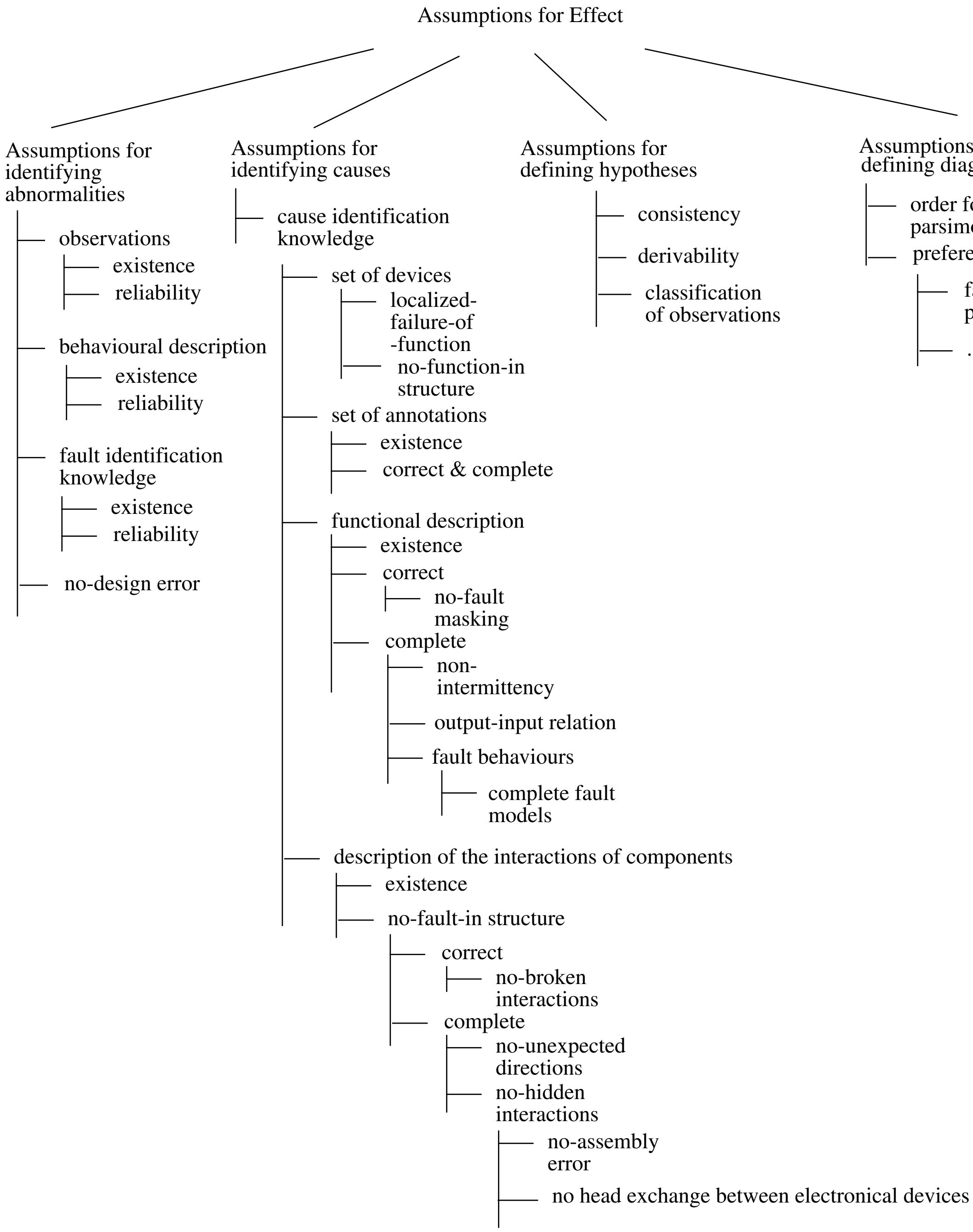

Fig. 1 Assumptions for Effect.

5. They can be weakened by representing all desired interactions as components (e.g., wires) that could fail; by representing additional possibilities of interactions (e.g., electronical devices can interact via heat exchange) [Böttcher, 1996]; by representing all potential unintended interaction paths between components [Preist & Welhalm, 1990]; by representing additional inputs to get rid of intermittency [Raiman et al., 1991].

3 The Problem Solver: Assumptions for Efficiency

Besides the assumptions that are necessary to define the task, further assumptions are necessary because of the complexity of model-based diagnosis. Component-based diagnosis is in the worst case exponential in the number of annotated components ([Bylander et al., 1991]). Every element of the power-set of the set of annotated components is a possible hypothesis. As we are not interested in problem-solving in principle but in practice, further assumptions have to be introduced that either decrease the worst-case, or at least the average-case behaviour.

3.1 Reducing the Worst-Case Complexity: The Single-Fault Assumption

A drastic way to reduce the complexity of the diagnostic task is achieved by the *single-fault* or *N-fault* assumption (SFA) [Davis, 1984], which reduces the complexity to polynomial in the number of components. If the single-fault assumption holds, the incorrect behaviour of the device is completely explainable by one failing component. This assumption defines either strong requirements on the provided domain knowledge, or significantly restricts the diagnostic problems that can correctly be handled by the diagnostic system (cf. [Benjamins et al., 1996]).

If the SFA has to be satisfied by the *domain knowledge*, then each possible fault has to be represented as a single entity. In principle this causes complexity problems for the domain knowledge as each fault combination (combination of fault components) has to be represented. However, additional domain knowledge can be used to restrict the exponential increase. [Davis, 1984] discusses an example of a representation change where a 4-fault case (i.e., 15 different combinations of faults) is transformed into a single fault. A chip with four ports can cause faults on each port. When we know that the individual ports never fail, but only the chip as a whole, a fault on four ports can be represented as one fault of the chip. Even without such a representation change, we do not necessarily have to represent all possible fault combinations. We can, for example, exclude all combinations that are not possible or likely in the specific domain (expert knowledge).

Instead of formulating the requirement above on the domain knowledge, one can also weaken the *task definition* by this assumption. This means that the competence of the PSM meets the task definition under the assumption that only single fault occurs. That is, only in cases where a single fault occurs, the method works correctly and complete. This would imply that the diagnostic system is designed for simple routine diagnoses.

3.2 Reducing the Average-Case Behaviour: The Minimality Assumption of GDE

As the SFA might be too strong an assumption for several applications, either as a requirement on the domain knowledge or as a restriction on the task, [Reiter, 1987] and [de Kleer & Williams, 1987] provide approaches able to deal with multiple faults. However, this re-introduces the complexity problems of MBD. To deal with this problem, GDE [de Kleer & Williams, 1987] exploits the *minimality assumption*, which reduces, in practical cases, the exponential worst case behaviour to a complexity that grows with the square of the number of components. In GDE, this assumptions helps reducing the complexity in two ways. First, a conflict is a set of components where it cannot be the case that all components work correctly given the provided domain knowledge and the observed behaviour. Under the minimality assumption, each super-set of a conflict is also a conflict and all conflicts can be represented by minimal conflicts. Second, a hypothesis contains at least one component of each conflict. Every super-set of such a hypothesis is again a hypothesis. Therefore, diagnoses can be represented by minimal diagnoses. The minimality assumption requires that diagnoses are independent or monotonic (see [Bylander et al., 1991]): a diagnosis that assumes more components as being faulty, explains more observations.

A drastic way to ensure that the minimality assumption holds, is to neglect any knowledge about the behaviour of faulty components. Thus, any behaviour that is not correct is considered as fault. A disadvantage of this is that physical rules may be violated (i.e., existing knowledge about faulty behaviour). We already mentioned the example provided in [Struss & Dressler, 1989], where a fault (one of two bulbs does not light) is explained by a broken battery that does not provide power and a broken bulb that lights without power. Knowledge about how components behave when they are faulty (called fault models) could be used to constrain the set of diagnoses derived by the system. On the other hand, it increases the complexity of the task. If for one component m possible fault behaviours are provided, this leads to $m+1$ possible states instead of two (correct and fault). The maximum number of candidates increases from 2^n to $(m+1)^n$.

A similar extension of GDE that includes fault models, is the Sherlock system (cf. [de Kleer & Williams, 1989]). With fault models, it is no longer guaranteed that every super-set of the faulty components that constitute the diagnosis, is also a diagnosis, and therefore the minimality assumption as such cannot be exploited. In Sherlock, a diagnosis does not only contain fault components (and implicitly assumes that all other, not mentioned, components are correct), but it contains a set of components assumed to work correctly and a set of components assumed to be fault. A conflict is now a set of some correct and fault components that is inconsistent with the provided domain knowledge and the observations. In order to accommodate to this situation, [de Kleer et al., 1992] extend the concept of minimal diagnoses to kernel diagnoses and characterise the conditions under which the minimality assumption still holds. The kernel diagnoses are given by the prime implicants of the minimal conflicts. Moreover, the minimal sets of kernel diagnoses sufficient to cover every diagnosis correspond to the irredundant sets of prime implicants[6] of all minimal conflicts. These extensions cause drastic additional effort, because there can be exponentially more kernel diagnoses than minimal diagnoses, and finding irredundant sets of prime implicants is NP-hard. Therefore, [de Kleer et al., 1992] characterise two assumptions under which the kernel diagnoses are identical to the minimal diagnoses. The kernel diagnoses are identical to the minimal diagnoses if all conflicts contain only fault components. In this case, there is again only one irredundant set of minimal diagnoses (the set containing all minimal diagnoses). The two assumptions that can ensure these properties are the *ignorance-of-abnormal-behaviour* assumption and the *limited-knowledge-of-abnormal-behaviour* assumption.

The ignorance-of-abnormal-behaviour assumption excludes knowledge about faulty behaviour and thus characterises the original situation of GDE. The limited-knowledge-of-abnormal-behaviour assumption states that the knowledge of abnormal behaviour does not rule out any diagnosis indicating a set of fault components, if there exist a valid diagnosis indicating a subset of them as faulty components, and if the additional assumed fault components are not inconsistent with the observations and the system description.[7] The latter assumption is a refinement of the former, that is, the truth of the ignorance-of-abnormal-behaviour assumption implies the truth of the limited-knowledge-of-abnormal-behaviour assumption.

A similar type of assumption is used by [Bylander et al., 1991] to characterise different complexity classes of component-based diagnosis. In general, finding one or all diagnoses is intractable. The *independent* and *monotonic* assumption, which have the same effect as the limited-knowledge-of-abnormal-behaviour assumption, require that each super-set of a diagnosis indicating a set of faulty components is also a diagnosis.[8] In this case, the worst-case complexity of finding one minimal diagnosis grows polynomially with the square of the number of components. However, the task of finding all minimal diagnoses is still NP-hard in the number of components. This corresponds to the fact that the minimality assumption of GDE

6. See [McCluskey, 1956].
7. [McIlraith, 1994] generalizes these assumptions for the dual case of diagnoses diagnosing a minimal set of components proven to be correct and applies them for characterizing minimal abductive diagnoses.
8. More precisely, the explanatory power of a hypothesis increases monotonously by adding fault or correct components.

(i.e., the ignorance-of-abnormal-behaviour and limited-knowledge-of-abnormal-behaviour assumptions), that searches for all diagnoses, does not change the worst-case but only the average-case behaviour of the diagnostic reasoner.

3.3 Search Guidance

The complexity of component-based diagnosis (especially when working with fault models) requires further assumptions that enable efficient reasoning for practical cases (cf. [Struss, 1992], [Böttcher & Dressler, 1994]). Again, these assumptions do not change the worst case complexity but should reduce the necessary effort in practical cases. A well-known notion to increase efficiency is a reasoning focus. Defining a focus for the reasoning process can be achieved by exploiting hierarchies or probability information. The *hierarchically-layered device-model* assumption assumes the existence of hierarchically layered models that allow step wise refinement of diagnosis to reduce the complexity of the diagnostic process (cf. the complexity analysis of hierarchical structures of [Goel et al., 1987]). The large number of components at the lowest level of refinement is replaced by a small number of components at a higher level. Only the relevant parts of the model are refined during the problem-solving process. The *hierarchically-layered behavioural-model* assumption assumes the existence of more abstract descriptions of the behaviour that can improve the efficiency because reasoning can be performed at a more coarse grained, and thus simpler, level (cf. [Abu-Hanna, 1994]). The *existence-of-probabilities* assumption assumes knowledge about the probability of faults that can be used to guide the search process for diagnoses by focusing on faults with high probabilities. Usually, these probabilities introduce new assumptions (e.g., the *components-fail-independently* assumption [de Kleer & Williams, 1989]).

All these knowledge types and their related assumptions rely on further assumptions concerning the utility of this search control knowledge. For example, the hierarchically-layered device model improves only the search process when the faults are not distributed in a way that enforces the problem solver to expand each abstract component descriptions to their lowest levels. It significantly improves the search process if the problem solver need to refine only one abstract component description at each level.

3.4 Summary

Figure 2 summarises the assumptions and groups them according to their purpose. All these assumptions are introduced to reduce the computational effort required to solve the problem. Table 2 provides an explanation of the assumptions along with the role they play (function), the domain they are about (case data, domain knowledge or task), and some references where they are discussed in more detail.

Table 2: Efficiency-Assumptions in component-oriented diagnosis.

name	explanation	is about[a]	function	some references
single fault (SFA), *N*-fault	There is one or there are at most *N* faults.	dk or t	It polynomializes the worst-case complexity for finding one or all diagnoses.	[Davis, 1984]
minimality	Sets of hypotheses can be represented by one minimal hypothesis.	dk	It polynomializes the average-case behaviour for finding all diagnoses.	[Reiter, 1987], [de Kleer & Williams, 1987], [Bylander et al., 1991], [de Kleer et al., 1992]
Ignorance-of-Abnormal-Behaviour	No knowledge that constrains possible faulty behaviour is provided.	dk	It polynomializes the average-case behaviour for finding all diagnoses.	[de Kleer & Williams, 1987], [de Kleer et al., 1992]

Table 2: Efficiency-Assumptions in component-oriented diagnosis.

name	explanation	is about[a]	function	some references
independency	The explanatory power of a diagnosis is the union of the explanatory power of its elements.	dk	It polynomializes the worst-case complexity for finding one diagnoses.	[Bylander et al., 1991]
monotonicity	The explanatory power of a diagnosis increases monotonously with its size.	dk	It polynomializes the worst-case complexity for finding one diagnoses.	[Bylander et al., 1991]
Limited-Knowledge-of-Abnormal-Behaviour	Valid diagnoses do not become invalid by adding further correct or fault components to it.	dk	It polynomializes the average-case behaviour for finding all diagnoses.	[de Kleer et al., 1992]
existence of search control knowledge	This knowledge is used to guide the search process for diagnoses.	dk	It improves the average-case behaviour for finding all diagnoses.	[Struss, 1992], [Böttcher & Dressler, 1994]
existence of a hierarchical-layered device-model	The device model is hierarchically structured.	dk	The hierarchical structure of the device focuses the search process.	[Goel et al., 1987], [Struss, 1992], [Böttcher & Dressler, 1994]
existence of a hierarchical-layered behavioural-model	The behavioural description of the system is hierarchically structured.	dk	Abstract descriptions of the behaviour should reduce the search effort.	[Struss, 1992], [Abu-Hanna, 1994], [Böttcher & Dressler, 1994]
existence of probabilities	Faults are annotated by their probability.	dk	Probabilities of faults focus the search process.	[de Kleer & Williams, 1987], [Struss, 1992], [Böttcher & Dressler, 1994]
fault probabilities are independent	Each fault appears independently from other possible faults.	cd & dk	It is used in computing probabilities for hypotheses.	[de Kleer & Williams, 1989]

a. cd = case data, dk = domain knowledge, t = task

Assumptions for Efficiency
- single or N-fault
- minimality
  - ingnorance of abnormal behaviour
  - independence of hypothes
  - monotonocity of hypotheses
  - limited-knowledge-of-abnormal behaviour
- search control knowledge
  - existence of a hierarchical-layered device model
  - existence of a hierarchical-layered behavioural model
  - existence of probabilities of hypotheses
    - independence of fault probabilities

Fig. 2 Assumptions for Efficiency.

4 Efficient Interaction with the Environment: Hypothesis Discrimination

Hypothesis *discrimination* becomes necessary if the number of hypotheses found, exceeds the desired number (cf. Table [Davis & Hamscher, 1988]). Additional observations must be provided as the initial observations were not strong enough to discriminate between existing hypotheses. Assumptions related to this activity include the following (see Table [Benjamins, 1993]). First, it must be possible to obtain *additional observations*. Examples of more specific versions of this assumption are: can the device be unfastened, are measuring points reachable, can components be replaced easily to test behaviour, can new input be provided to the device, etc. Second, assumptions can be made about the *utility of additional observations*. One can assume cost information of additional measurements and knowledge about their discriminatory power (i.e., knowledge about dependencies between hypotheses) to optimise their selection. Again, NP-hard problems arise if one tries to optimise these decisions. Therefore, assumptions concerning *heuristic knowledge* that guide this process are necessary.

Table 3: Efficiency-Assumptions in component-oriented diagnosis.

name	explanation	is about[a]	function	some references
Possibility of additional observations	What are further possible observations.	dk	It is necessary to get further information for hypotheses discrimination.	[Davis & Hamscher, 1988], [Benjamins, 1993]
Utility of additional observations	How useful are these further observations (information gain versus costs).	dk	It necessary for optimal decisions during hypotheses discrimination.	[de Kleer & Williams, 1987], [Davis & Hamscher, 1988], [Benjamins, 1993]
heuristic search knowledge	This knowledge is used to guide the search process for optimal selection of further observations.	dk	It necessary for efficiently making sub-optimal decisions.	[de Kleer & Williams, 1987], [Davis & Hamscher, 1988], [Benjamins, 1993]

a. cd = case data, dk = domain knowledge, t = task

All these assumptions are necessary to optimise the cooperation of the diagnostic system with its environment. In principle, one could assume that all observations that are possible are provided to the system before it starts its diagnostic reasoning. However, collecting observations is often a major cost-determining factor. Therefore, assumptions are introduced concerning the efficiency of gathering information with minimal costs.

5 The Role of Assumptions in Specifying Knowledge-Based Reasoning Systems

In [Fensel et al., 1996], we provided different aspects of a specification of knowledge-based system which are related by assumptions (see Figure 3): a *task definition* defines the problem to be solved by the knowledge-based system; a *problem-solving method* defines the reasoning process of the knowledge-based system; and a *domain model* describes the domain knowledge of the knowledge-based system.[9]

The *task definition* specifies the goals that should be achieved in order to solve a given problem, which are functionally specified as an input-output relation. A task definition also defines assumptions about the domain knowledge. For example, a task that concerns the selection of the maximal element of a set of elements, requires a preference relation as domain knowledge. Assumptions are also used to define the requirements on such a relation (e.g. transitivity, symmetry, etc.).

The reasoning of a knowledge-based system can be described by a *problem-solving method* (*PSM*). A PSM consists of three parts. First, a definition of the functionality defines the *competence* of the PSM independent of its realisation. Second, an *operational description* defines the dynamic reasoning process. Such an operational description describes how the competence can be achieved in terms of the reasoning steps and their dynamic interaction (i.e., the knowledge and control flow). The third part of a PSM concerns *assumptions* about the domain knowledge. Each inference step requires a specific type of domain knowledge with specific characteristics. These complex requirements on the input of a problem-solving method distinguish it from usual software products. Preconditions on inputs (used in normal software to guarantee valid inputs) are in PSMs extended to complex requirements on available domain knowledge. A problem-solving method can only solve a complex task with reasonable computational effort by introducing assumptions. Such assumptions can work in two directions to achieve this result. First, they can restrict the complexity of the problem, that is, weaken the task definition in such a way that the PSM competence is sufficient to realise the task. Second, they can strengthen the competence of the PSM by the assumed (extra) domain knowledge. This is called the *law of conservation of assumptions* in Figure [Benjamins et al., 1996]. In terms of dynamic logic Figure [Harel, 1984], this law can be formulated as follows:

$$assumptions_{domain\text{-}knowledge} \rightarrow ([problem\text{-}solving\ method]\ assumptions_{task} \rightarrow goal)$$

The formula states that, if the assumptions on domain knowledge hold in the initial state, then the problem-solving method must terminate in a state that fulfils the goals of the task, if the assumptions on the task are fulfilled. This formula could be weakened by either strengthening the assumptions on domain knowledge (i.e., the problem-solving method must behave well for less initial states) or strengthening the assumptions on the task (i.e., the problem-solving method must achieve the goal in less terminal states).

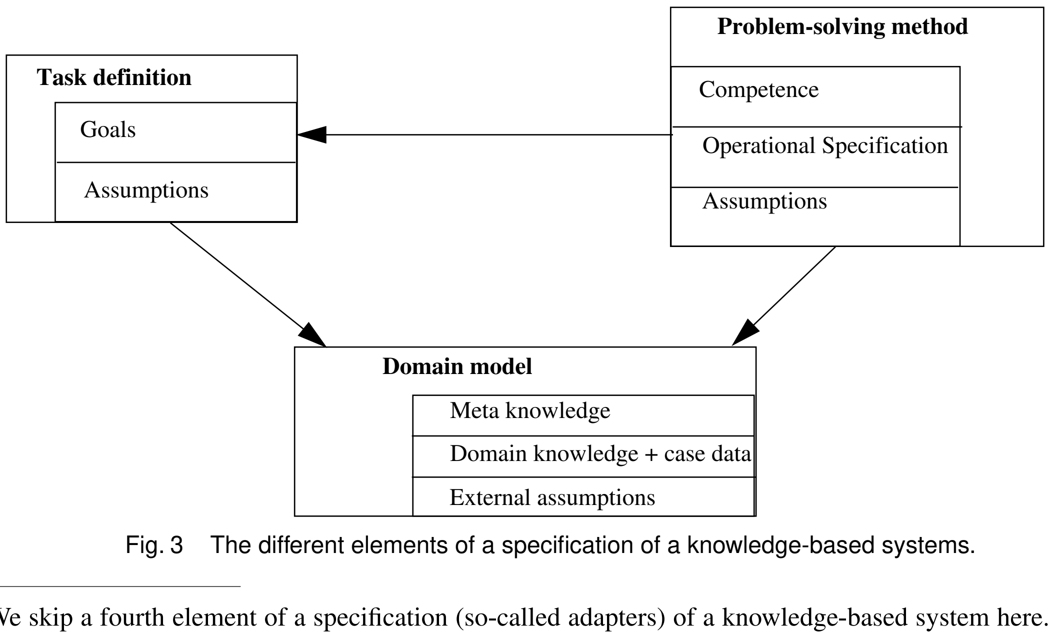

Fig. 3 The different elements of a specification of a knowledge-based systems.

9. We skip a fourth element of a specification (so-called adapters) of a knowledge-based system here. The necessity of adapters arise if the other buildings blocks should be reusable. Then they must be adjusted to each other and to the specific requirements of a given application problem.

The description of the *domain model* introduces the domain knowledge as it is required by the problem-solving method and the task definition. Three elements are needed to define a domain model. First, a description of properties of the domain knowledge at a meta-level. The *meta-knowledge* characterises properties of the domain knowledge. It is the counterpart of the assumptions on domain knowledge made by the other parts of a KBS specification: assumptions made about domain knowledge by these parts, must be stated as properties of the domain knowledge. These properties define goals for the modelling process of domain knowledge in the case of knowledge acquisition. The second element of a domain model concerns the *domain knowledge* and *case data* necessary to define the task in the given application domain, and necessary to carry out the inference steps of the chosen problem-solving method. The third element is formed by *external assumptions* that link the domain knowledge with the actual domain. These external assumptions can be viewed as the missing pieces in the proof that the domain knowledge fulfils its meta-level characterisations.

The assumptions of the different parts of a KBS specification define their proper relationships and the adequate relationship of the overall specification with its environment. Their detection, verification, and validation is an important part for developing correct reasoning systems. In [Fensel et al., 1996], we used the Karlsruhe Interactive Verifier (KIV)[10] [Reif, 1995], developed in the specifications. Besides verification, KIV was used to detect hidden assumptions which were necessary to relate the competence of a problem-solving method to the task definition. Hidden assumptions could be found by analysing failed proof attempts. The analysis of partial proofs gives some hints for the construction of possible counter examples, but also for repairing the proof by introducing further assumptions. These assumptions are the *missing pieces* in proofing the correctness of the specification. Verifying these specification is therefore a way to detect underlying hidden assumptions.

6 Conclusion

It is essential to know the underlying assumptions of a reasoning system in order to know when it is applicable. Moreover, assumptions are good ways to characterise them and they can be used to guide the acquisition process of domain knowledge. They define the type of knowledge and its properties as it is are required by the reasoner.

In [Fensel, 1995] we provided the analysis of the assumptions of one problem-solving method. In this paper we have provided a study of the role of assumptions used by a the group of problem-solving methods for model-based diagnosis. Model-based diagnostic systems were first introduced to overcome the limitations of heuristic systems. However, research on model-based systems revealed immediately that “pure“ model-based diagnostic systems were too inefficient to be useful. Model-based diagnosis is in principle intractable, and in the last ten years, substantial effort has been devoted to optimise diagnostic algorithms in order to make them tractable. Assumptions can be viewed as the re-introduction, however controlled, of heuristics for reasons of efficiency. Looking at the history of model-based diagnostic systems in retrospect, we could describe it as follows: researchers have started with a general, but inefficient, model-based diagnostic reasoner, and then incrementally introduced and modified assumptions, until an efficient reasoner was achieved. This approach is strikingly similar to the assumption-driven development process of problem solvers recently put forward in the knowledge engineering community [Fensel & Straatman, 1996], where one starts with a general, but inefficient, specification of the reasoning process and gradually introduces and modifies assumptions until a efficient reasoner is achieved. In [Fensel et al., 1996] is shown how verification tools can be used to support the process of detecting and introducing assumptions of reasoning systems.

---

10. The KIV system (Karlsruhe Interactive Verifier) is an advanced tool for the construction of provably correct software. It supports the entire design process starting from formal specifications and ending with verified code.

In future work we will use the assumptions identified in this paper to construct, in an assumption-driven manner, efficient knowledge-based reasoners. We will also investigate whether the identified assumptions can be structured in a such a way that they support this process. In other words, we are looking for libraries of reusable assumptions. Still, our current work suffers from serious shortcomings. We have not provided a proper definition of the term *assumption* nor a real justification for the classification we applied in the paper. The assumptions where collected as they could be found in papers on model-based diagnosis but it lacks a theoretical-based framework that would allow to detect gaps in this list and to decide when a level of (relative) completeness is achieved.

Acknowledgement. We would like to thank Annette ten Teije for guiding us through the literature on model-based diagnosis and for helpful comments on early drafts of the paper. Helpful comments were also provided by Claudia Böttcher, Bert Bredeweg, Joost Breuker, Kees de Koning, Remco Straatman, and anonymous reviewers.

References

[Abu-Hanna, 1994] A. Abu-Hanna: Multiple Domain Models in Diagnostic Reasoning, PhD thesis, University of Amsterdam, 1994.

[Benjamins, 1993] R. Benjamins: Problem-Solving Methods for Diagnosis, PhD Thesis, University of Amsterdam, Amsterdam, the Netherlands, 1993.

[Benjamins et al., 1996] R. Benjamins, D. Fensel, and R. Straatman: Assumptions of Problem-Solving Methods and Their Role in Knowledge Engineering. In *Proceedings of the 12. European Conference on Artificial Intelligence (ECAI-96)*, Budapest, August 12-16, 1996.

[Benjamins & Pierret-Golbreich, 1996] R. Benjamins and C. Pierret-Golbreich: Assumptions of Problem-Solving Method. In N. Shadbolt et al. (eds.), *Advances in Knowledge Acquisition*, Lecture Notes in Artificial Intelligence (LNAI), no 1076, Springer-Verlag, Berlin, 1996.

[Böttcher, 1996] C. Böttcher: No Faults in Structure? - How to Diagnose Hidden Interactions, 1996.

[Böttcher & Dressler, 1994] C. Böttcher and O. Dressler: A Framework For Controlling Model-Based Diagnosis Systems with Multiple Actions, *Annals of Mathematics and Artificial Intelligence*, 11:241-261, 1994.

[Bylander, 1991] T. Bylander: Complexity Results for Planning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, August 1991.

[Bylander et al., 1991] T. Bylander, D. Allemang, M. C. Tanner, and J. R. Josephson: The Computational Complexity of Abduction, *Artificial Intelligence*, 49: 25-60, 1991.

[Chandrasekaran, 1991] B. Chandrasekaran: Models versus rules, deep versus compiled, content versus form, *IEEE-Expert*, 6(2): 75--79.

[Console & Torasso, 1992] L. Console and P. Torasso: A Spectrum of Logical Definitions of Model-Based Diagnosis. In W. Hamscher et al. (eds.), *Readings in Model-based Diagnosis*, Morgan Kaufman Publ., San Mateo, CA, 1992.

[Davis, 1984] R. Davis: Diagnostic Reasoning Based on Structure and Behavior, *Artificial Intelligence*, 24: 347-410, 1984.

[Davis & Hamscher, 1988] R. Davis and W. Hamscher: Model-based Reasoning: Troubleshooting. In H. E. Shrobe (ed.), *Exploring AI: Survey Talks from the National Conference on AI*, Morgen Kaufman, San Mateo, CA, 1988.

[Fensel, 1995] D. Fensel: Assumptions and Limitations of a Problem-Solving Method: A Case Study. In *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW-95)*, Banff, Canada, February 26th - February 3th, 1995.

[Fensel et al., 1996] D. Fensel, A. Schönegge, R. Groenboom and B. Wielinga: Specification and Verification of Knowledge-Based Systems. In *Proceedings of the ECAI-96 Workshop Validation, Verification and Refinement of Knowledge-Based Systems, at the 12th European Conference on Artificial Intelligence (ECAI-96)*, Budapest, 12.-16. August 1996. See also: *Proceedings of the 10th*

Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW-96), Banff, Canada, November 9-14, 1996

[Fensel & Straatman, 1996] D. Fensel und R. Straatman: The Essence of Problem-Solving Methods: Making Assumptions for Efficiency Reasons. In N. Shadbolt et al. (eds.), *Advances in Knowledge Acquisition*, Lecture Notes in Artificial Intelligence (LNAI), no 1076, Springer-Verlag, Berlin, 1996.

[Genesereth, 1984] M. R. Genesereth: The Use of Design Descriptions in Automated Diagnosis, *Artificial Intelligence (AI)*, 24:411-436, 1984.

[Goel et al., 1987] A. Goel, N. Soundararajan, and B. Chandrasekaran: Complexity in Classificatory Reasoning. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87)*, Seattle, Washington, July 13-17, 1987.

[Harel, 1984] D. Harel: Dynamic Logic. In D. Gabby et al. (eds.), *Handbook of Philosophical Logic, vol. II*, Extensions of Classical Logic, Publishing Company, Dordrecht (NL), 1984.

[de Kleer & Brown, 1984] J. de Kleer and J. S. Brown: A Qualitative Physics Based on Confluences, *Artificial Intelligence*, 24:7-83, 1984.

[de Kleer et al., 1992] J. de Kleer, A. K. Mackworth, and R. Reiter: Characterizing Diagnoses and Systems, *Artificial Intelligence*, 56, 1992.

[de Kleer & Williams, 1987] J. de Kleer and B. C. Williams: Diagnosing Multiple Faults, *Artificial Intelligence*, 32:97-130, 1987.

[de Kleer & Williams, 1989] J. de Kleer and B. C. Williams: Diagnosis with Behavioral Modes. In *Proceedings of the 11th International Joint Conference on AI (IJCAI-89)*, Detroit, MI, 1989.

[McCluskey, 1956] E. J. McCluskey: Minimizing of Boolean Functions, *Bell Systems Technology Journal*, 35(5):1417-1444, 1956.

[McIlraith, 1994] S. McIlraith: Further Contribution to Characterizing Diagnosis, *Annals of Mathematics and AI*, special issues on model-based diagnosis, 11(1-4), 1994.

[Nebel, 1996] B. Nebel: Artificial intelligence: A Computational Perspective. To appear in G. Brewka (ed.), *Essentials in Knowledge Representation*.

[Nejdl et al., 1995] W. Nejdl, P. Froehlich and M. Schroeder: A Formal Framework For Representing Diagnosis Strategies in Model-Based Diagnosis Systems. In *Proceedings of the 14th International Joint Conference on AI (IJCAI-95)*, Montreal, Canada, August 20-25, 1995.

[Preist & Welhalm, 1990] C. Preist and B. Welham: Modelling Bridge Faults fo Diagnosis in Electronic Circuits. In *Proceedings of the 1st International Workshop on Principles of Diagnosis*, Stanford, 1990.

[Raiman, 1989] O. Raiman: Diagnosis as a Trial. In *Proceedings of the Model Based Diagnosis International Workshop*, Paris, 1989.

[Raiman, 1992] O. Raiman: The Alibi Principle. In W. Hamscher et al. (eds.), Readings in model-based diagnosis, Morgan Kaufmann Publ., San Mateo, CA, 1992.

[Raiman et al., 1991] O. Raiman, J. de Kleer, V. Saraswat, M. Shirley: Characterizing Non-Intermittent Faults. In *Proceedings of the 9th National Conference on AI (AAAI-91)*, Anaheim, CA, July 14-19, 1991.

[Reif, 1995] W. Reif: The KIV Approach to Software Engineering. In M. Broy and S. Jähnichen (eds.): *Methods, Languages, and Tools for the Construction of Correct Software*, Lecture Notes in Computer Science (LNCS), no 1009, Springer-Verlag, Berlin, 1995.

[Reiter, 1987] R. Reiter: A Theory of Diagnosis from First Principles, *Artificial Intelligence*, 32:57-95,1987.

[Sticklen et al., 1989] J. Sticklen, B. Chandrasekaran, and W.E. Bond: Applying a Functional Approach for Model Based Reasoning, Proc. of (IJCAI) Workshop on Model Based Reasoning, Detroit, 1989,

[Struss, 1992] P. Struss: Diagnosis as a Process. In W. Hamscher etal. (eds.), *Readings in Model-based Diagnosis,* Morgan Kaufman Publ., San Mateo, CA, 1992.

[Struss & Dressler, 1989] P. Struss and O. Dressler: “Physical Negation“ - Integrating Fault Models into the General Diagnostic Engine. In *Proceedings of the 11th International Joint Conference on AI (IJCAI-89)*, Detroit, MI, 1989.

[ten Teije & van Harmelen, 1996] A. ten Teije and F. Van Harmelen: Using reflection techniques for flexible problem solving (with examples from diagnosis), *Future Generation Computer Systems*, to appear 1996.