A Formal Semantics for Preferences and Strategies in Model-Based Diagnosis

Peter Fröhlich, Wolfgang Nejdl, Michael Schroeder
Lehrstuhl für Informatik V
RWTH Aachen
52062 Aachen, Germany
{froehlich,nejdl,schroeder}@informatik.rwth-aachen.de

Abstract

The concept of diagnosis as a process has been coined to introduce a dynamic representation of the diagnosis process and subsequent work has focused on formalizing this notion. Advancing on previous proposals built upon rather procedural semantics, we aim to give a formal and declarative semantics for the diagnosis process. We give a logical characterization of preferences and strategies, expressed by preference relations on single diagnoses and modal logic formulas on sets of diagnoses, respectively.

1 Introduction

Overview

In most diagnosis systems, diagnosis is considered as a static problem, i.e. the system description and the diagnosis goals are not changed during the diagnosis process. As already observed in recent work such as [10] and [1, 2], this is a big disadvantage for the diagnosis of complex systems. All informations such as different specifications, modeling assumptions, abstractions etc. are visible during the whole diagnosis process.

Therefore the concept of diagnosis as a process has been coined in [10] and [11] to introduce a dynamic representation of the diagnosis process. Subsequent work has focused on formalizing this notion, most of them focusing on specific features of this process. Friedrich covers the principle of abstraction in [7], Damasio, Nejdl and Pereira focus on preference relations for diagnoses [5]. The most general approach up to now has been defined by Böttcher and Dressler in [1, 2], including a set of preferences/strategies to guide the diagnosis process. Although they provide a formal definition for preferred diagnosis [6] their description and semantics for diagnosis strategies is rather intertwined with the workings of an ATMS-based diagnosis engine, resulting in rather procedural semantics explicitly based on iteration. They show how to compute diagnoses, but cannot characterize their results.

So while the theory of model-based diagnosis is well in place, a declarative theory of using preferences and strategies during the diagnosis process independent of specific systems is still missing. Our aim in this paper is to provide a first realization of such a theory, based on preference based semantics for reasoning with preferences and on modal logic strategy formulas for capturing strategic decisions about working hypotheses.

Preferred Diagnoses

The first formal characterization of model-based diagnosis [9] introduced the concept of set-theoretically minimal diagnoses to describe which consistent diagnosis hypothesis are preferred to others. Subsequent systems usually used a preference relation based on probabilities. These approaches are all special cases of a general approach, which ascribes to each (consistent) diagnosis hypothesis a set of properties, and specifies how diagnoses with certain properties are preferred to diagnoses with certain other properties. For example a diagnosis containing a single fault might be considered better than one with more than one fault. In this paper we will define such a generalized preference concept based on properties of single diagnoses.

Strategies

Another concept is needed to model strategic decisions during the diagnosis process. There we need to evaluate a set of preferred diagnoses, and continue diagnosis under a different set of working hypotheses depending on the properties of this set of diagnoses. For example, if all preferred diagnoses accuse a component of being faulty, we might want to consider refining the model for this component, producing a set of more specific diagnoses involving subcomponents of the suspected component. We will call the sequence of sets of working hypotheses adopted during the diagnosis process a *strategy*. Thus a strategy describes the process of finding a satisfactory set of diagnoses. Strategic knowledge (i.e. which working hypotheses are adopted dependent on the properties of the current set of diagnoses) is expressed in modal logic on a connected directed graph of $S5$–structures.

2 Preference Graph

2.1 Starting Point

We will use a first order language with equality (called $\mathcal{L}_{Obj}$) as general background. $ATOMS$ denotes the set of atoms of $\mathcal{L}_{Obj}$, LIT is the set of literals, i.e. positive and negative atoms. The concept of preferences during the diagnosis process is based on the diagnosis framework as introduced in [9], where diagnoses were defined in the following way:

Definition 2.1 *Minimal Diagnoses*
Let SD, OBS and $COMP$ denote the system description, the observations and the components, respectively. A *minimal diagnosis* for $SD \cup OBS$ is a minimal set $\Delta \subseteq COMP$ such that

$$SD \cup OBS \cup \{ab(c) | c \in \Delta\} \cup \{\neg ab(c) | c \in COMP - \Delta\}$$

is consistent.

In the above definition a component's ok–mode is preferred to abnormality. This is the usual approach, which we will assume, too, without loss of generality. The preference graph defined in the following sections can easily be modified for the case where abnormality is preferred over non-abnormality.

2.2 Properties of Diagnoses

To calculate minimal diagnoses we have to find all minimal models (with respect to ab) of the theory $SD \cup OBS$. In cases where we have a large set of minimal models, we want to refine the notion of preferred model by defining preference relations between minimal models. Minimality as stated in the definition of minimal diagnoses uses set inclusion to define preference. This is unsatisfactory as we may want to express that a solution is minimal in terms of size or preferred because of some properties it has. Therefore we define a preference graph that allows the user to refine the notion of preference.

We express the preference of one minimal model to another in terms of properties. One model may satisfy a given property while another does not. These properties are ordered in a directed preference graph which describes which properties are preferred to others, producing a partial ordering of all possible minimal models. The extended definition of a preferred diagnosis demands not only minimality with respect to ab, but also minimality with respect to the partial ordering induced by the preference graph.

We will use *properties* as an indicator whether a diagnosis satisfies a certain condition. A property is expressed in terms of a literal.

Definition 2.2 *Property literals, Property rules*
Let $PROP \subseteq LIT$ be a set of literals, then $p \in PROP$ is called property. For each property $p \in PROP$ the system description SD may contain a property rule $p \leftrightarrow c$, where c is a formula of the language.

Example 2.1
In order to express the number of faulty components a diagnosis contains, we define the property literals $single\_faults$ and $double\_faults$. The system description contains the property rules

$$\begin{aligned} &single\_faults \leftrightarrow \\ &(\forall X : \forall Y : ab(X) \wedge ab(Y) \rightarrow X = Y) \\ &double\_faults \leftrightarrow \\ &(\forall X : \forall Y : \forall Z : ab(X) \wedge ab(Y) \wedge ab(Z) \rightarrow \\ &\qquad X = Y \vee X = Z \vee Y = Z) \end{aligned}$$

With the property rules we are able to relate a property to a diagnosis. A minimal model of $SD \cup OBS$, where SD contains property rules, indicates which properties are satisfied by the underlying diagnosis. In order to enforce a single property p, we add the property literal denoting the desired property to the system description. This ensures that a minimal diagnosis for the new system description satisfies this property, i.e. a minimal diagnosis for $SD \cup \{p\} \cup OBS$ guarantees the property denoted by the property literal p.

By this method we are able to obtain minimal diagnoses satisfying a *single* property. In order to express more general satisfiability conditions, we allow the addition of a formula of $\mathcal{L}_{Obj}$ based on property literals. Such a formula is called property constraint.

Definition 2.3 *Property constraint*
A formula pc of $Lang$ being composed of property literals, truth values $true$ and $false$, connectors $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$ and quantifiers $\forall$, $\exists$ is a property constraint. Nothing else is a property constraint.

Example 2.2 *(continued)*
Two property constraints are

$$single\_faults \wedge \neg double\_faults$$

and

$$single\_faults \vee double\_faults$$

We express that the former is preferable to the latter using a preference graph.

2.3 Preferred Diagnosis

With property rules and property constraints we are able to check and enforce properties of diagnoses. Given a preference relation over the property constraints, a minimal model being also minimal with respect to this preference relation is preferred to non-minimal ones. The following definition of preference graph is an extension of the one given in [5].

Definition 2.4 *Preference Graph*
A directed, connected, acyclic graph $G = (V, E)$ is called a preference graph referring to a system description SD and a set of property literals $PROP$, iff

- V is a set of nodes and E a set of edges,
- each node $v \in V$ has a unique label and a property constraint over $PROP$ and
- there is a node labeled $-$ (bottom) which is not accessible by the other nodes, while all other nodes are accessible from $-$.

Sometimes we do not distinguish between a node and its label or its property constraint.

A node v is preferred to a node w, iff w is reachable from v, while v is not reachable from w. As no other nodes have access to $-$ it is the smallest node.

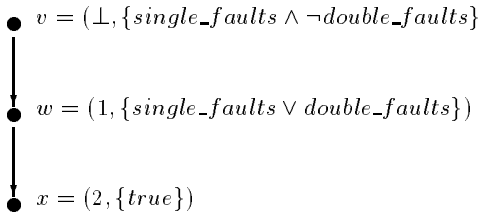
Definition 2.5 *Minimal Node*
Let $G = (V, E)$ be a preference graph and $v, w \in V$, then w is reachable by v, iff there is a path

$$\{(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)\}, \quad (v_i, v_{i+1}) \in E,$$

such that $v_0 = v$, $v_k = w$.

$v < w$ holds, iff w is reachable by v, whereas v is not reachable by w. $v \in V$ is minimal, iff $\nexists w \in V. w < v$.

Example 2.3 *(continued)*
A preference graph to prefer single to double to any number of faults has the form $G = (\{v, w, x\}, \{(v, w), (w, x)\}$ with nodes as given below. $v < w$ and $w < x$ hold.



With the preference graph all concepts of minimality are covered. Diagnoses need not be minimal with respect to set inclusion only, but they have to be minimal with respect to the preference graph, too. This gives us the following definition of preferred diagnosis:

Definition 2.6 *Preferred Diagnosis*
Let SD, OBS, $COMP$ and $G = (V, E)$ denote the system description, the observations, the components and the preference graph, respectively. A *preferred diagnosis* for $SD \cup OBS$ and G is a minimal set $\Delta \subseteq COMP$, such that

$$SD \cup \{pc\} \cup OBS$$
$$\cup \quad \{ab(c) | c \in \Delta\} \cup \{\neg ab(c) | c \in COMP - \Delta\}$$

is consistent, where pc is the property constraint of a node $v \in G$, and there is no $v_1 \in G$ with property constraint pc_1, such that $v_1 < v$ and

$$SD \cup \{pc_1\} \cup OBS$$
$$\cup \quad \{ab(c) | c \in \Delta\} \cup \{\neg ab(c) | c \in COMP - \Delta\}$$

is consistent.

2.4 Sample Property Rules

In the following paragraphs we discuss some property rules. faults_complete, n_faults are adopted from [1, 2].

- **Choice of specification.**
  If there are different ways to specify the problem, we want to switch between these specifications (views). Given n specifications of the examined system. All rules belonging to specification I contain the property literal $spec(I)$ in their body. The property rule

  $$spec_i \leftrightarrow \bigwedge_{j=1, j\neg i}^{n} \neg spec_j$$

  states that only one problem specification is active at a time. So rules belonging to other specifications are faded out and are only considered during the correct phase of the diagnosis process.
- **Number of faults.**
  If $n < m$ holds for integers n and m a diagnosis containing n faulty components is preferable to one with m. The diagnosis contains n components, if exactly two of the $n + 1$ Variables $X_1 \ldots X_{n+1}$, who are bound to faulty components, are the same. If n_faults denotes the property literal, $\bigvee$ is an exclusive or of arity $n + 1$ that is true if exactly one of its members is true, while all others are false.

  $$n\_faults \leftrightarrow$$
  $$\forall_{i=1}^{n+1} X_i : \quad \bigwedge_{i=1}^{n+1} ab(X_i) \rightarrow \bigvee_{i,j=1, i\neq j}^{n+1} X_i = X_j$$

  If SD contains property rules for 1 to n faults, the property constraint

  $$i\_faults \wedge \bigwedge_{j=1, j\neq i}^{n} \neg j\_faults$$

  expresses that exactly i faults are enforced, while

  $$\bigvee_{j=1}^{i} i\_faults \wedge \bigwedge_{j=i+1, j\neq i}^{n} \neg j\_faults$$

  enforces the number of faulty components to be less or equal to i.
- **Preference of Fault modes.**
  For a given component c with possible fault modes $fm_1(c)$ to $fm_n(c)$ the fault mode $fm_i(c)$ may be the most plausible one. The property rule

  $$only\_fm_i(c) \leftrightarrow \bigwedge_{j=1, j\neq i}^{n} \neg fm_j(c)$$

  states that other modes than i are not possible by the property constraint $only\_fm_i$.

- **Faults Complete**
  If the fault modes of a component are assumed to be complete the unknown fault mode is forbidden.

  $$faults\_complete(c) \leftrightarrow \neg fm_{unknown}(c)$$

  If the property constraint contains the property literal $faults\_complete$ this property is enforced.

With these properties and many others the user can define a preference graph as needed. Note, if we have a lot of partial preferences which are independent of each other (such as in the case of a local preference order on fault modes for each component), the combination of all of these independent preferences can lead to a large preference graph. However, this does not concern the user, as such a combination can be done automatically. Additionally, the preference graph can be constructed in a lazy fashion, building only the nodes as needed and forgetting the ones which are not needed any more. This is also true for encoding probability based preferences. Therefore, the theoretical worst case size of the preference graph is of no practical significance. In the examples we have done so far, the size of the preference graph is negligible to the size of the theory describing the system to be diagnosed.

3 Strategies

3.1 Motivation

Property rules refer to one diagnosis. Such properties are easy to code, as they can be expressed as part of the system description. This changes when a property depends on all possible diagnoses. In order to check such properties we already have to have a given set of diagnoses.

Diagnosis agents influence the diagnosis process by means of working hypotheses. For example, the agent may choose to view some component c at a more detailed level. This can be expressed by the working hypothesis $refine(c)$. When this hypothesis is asserted is specified in a strategy formula and depends on a property of the current set of diagnoses. From now on We call this properties *conditions*, to distinguish them from the properties used in the preference graph.

Example 3.1 *Structural Refinement*
In order to increase the efficiency of the diagnostic process hierarchies are introduced in the system description. In the beginning of the diagnostic process only the most abstract model of the system is active. While diagnosis proceeds, more detailed models of abnormal system components are considered. The process of refining the models is guided by the following rule:

> If an abstract system component C occurs in all diagnoses, we prefer to activate a more detailed model for C.

The condition above depends on a given set of diagnoses. In general, however, we need the complete models gained from $SD \cup OBS$ and the involved diagnoses, such as in the case of evaluating the utility of observations. Then we can express for example that a certain measurement has to be made if there are two models predicting contradictory values for that observation.

3.2 Syntax for Strategy Formulas

We assume that the system description SD and the set of observations OBS are expressed in the language $\mathcal{L}_{Obj}$. For simplicity, we interpret all atoms of $\mathcal{L}_{Obj}$ as propositions. However the semantics presented here can be extended to handle quantifiers as well. We already stated that the agent uses certain working hypotheses in order to influence the set of diagnoses.

Definition 3.1 *Working Hypothesis, Strategy Set*
Let $WHYP \subseteq LIT$ be a set of literals suited for describing the effect of strategies. A literal $L \in WHYP$ is called a *working hypothesis*. A set of working hypotheses is called a *Strategy Set*.

The working hypotheses the agent assumes depend on the properties of a given set of diagnoses. These properties can be expressed by means of strategy conditions.

Definition 3.2 *Strategy Condition*

1. Let F be a formula of the Object Level $F \in \mathcal{L}_{Obj}$. Then $\Box F$ and $\Diamond F$ are Strategy Conditions.
2. Let C, C' be Strategy Conditions. Then for $\circ \in \{\wedge, \vee, \rightarrow\}$ $C \circ C'$ is a Strategy Condition and $\neg C$ is a Strategy Condition. Similarly, $\Box C$ and $\Diamond C$ are Strategy Conditions.

The language $\mathcal{L}_{Cond}$ thus defined is similar to an ordinary modal language, with the additional requirement, that all formulas have to be in the scope of either $\Box$ or $\Diamond$, as strategy formulas are always interpreted on a set of preferred diagnoses and are not attached to a specific diagnosis.

A strategy condition is a statement about a set of diagnoses. As we will see later, this set of diagnoses corresponds to a strategy set currently assumed by the agent.

Example 3.2 *Strategy Condition*
Consider a system description where $ab(c)$ denotes that component c is abnormal and $val(x, p, y)$ denotes that the port p of component x has the value y. Then

$$\Box ab(c) \wedge \Diamond val(c, in1, 0)$$

is a strategy condition. The intended semantics of this condition is: The agent knows that $ab(c)$ occurs in all his current diagnoses. Furthermore, there is at least one diagnosis predicting a value of 0 for the first input of c.

If the set of diagnoses satisfies a certain strategy condition, the agent can assume a working hypotheses suitable for that situation. This connection between strategy condition and working hypotheses is modelled by strategy formulas.

Definition 3.3 *Strategy Formula*
Let C be a strategy condition and $L_1, \ldots L_n \in WHYP$. Then

$$C \rightarrow \Diamond\Box L_1 \vee \ldots \Diamond\Box L_n$$

is a Strategy Formula. The Language consisting of all strategy formulas is called $\mathcal{L}_{Strat}$ in the remainder of this paper.

Intuitively $\Diamond L$ means that L is a supported working hypotheses for integration into the next strategy set of the diagnosis agent. The rather restricted language presented so far is sufficient for the specification of all strategies presented in this paper. Most of the strategies have the form $C \rightarrow \Diamond L$. A disjunction of strategy literals can be used to express dependencies between strategies (see subsection 3.6).

Example 3.3 *Structural Refinement*
The strategy formula for structural refinement is

$$\Box ab(c) \rightarrow \Diamond\Box refine(c)$$

3.3 Semantics for strategy formulas

In this section we define a declarative semantics for the process of strategy evaluation based on modal logics evaluated on connected directed graph of $S5$–structures, where each $S5$–cluster corresponds to a set of diagnoses under a given strategy set, and a path through this graph corresponds to a strategy. A semantics for strategies can be judged by the following criteria:

C1: (Logical Consistency) A strategy has to be logically consistent, i.e. it must lead to consistent models of the system.

C2: (Preferredness) A strategy must result in an $S5$–cluster, where all necessary working hypotheses suggested by the strategy formulas have been adopted.

C3: (Minimality) Only working hypotheses suggested by strategy formulas should be adopted.

We are going to define a strategy class, which can explicitly cope with C1 and C2. However, there is a tradeoff between C3 and efficient implementation. So, we are going to introduce a strategy concept that only satisfies a weak minimality concept (Local Minimality) but can be implemented efficiently.

We start defining our semantics by giving a meaning to the strategy conditions. These conditions are interpreted by standard S5-Models [3]. We construct an S5–Model from a strategy set A in the following manner:

Definition 3.4 *Induced S5–Model M_A*
Let A be a strategy set. Then we define the *induced S5–model* $M_A := \langle W, R, P\rangle$, where

1. W is a set of identifiers corresponding to all preferred diagnoses of $SD \cup OBS \cup A$
2. $R = W \times W$
3. P is a mapping $ATOMS \rightarrow W$, assigning to each atom a (possibly empty) set of preferred diagnoses in which it occurs.

The set of preferred diagnoses is used here in the sense as defined in Section 2. Note, that in general the mapping has to define the truth value not only of abnormality predicates, but all other predicates used in the strategy formulas.

Strategy conditions are evaluated as follows:

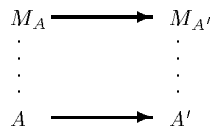
Definition 3.5 *Semantics of $\mathcal{L}_{Cond}$*
Let $M = \langle W, R, P\rangle$, be an $\mathcal{L}_{Cond}$–Model, where $W \neq \emptyset$. F, F_1, F_2 are strategy conditions:

1. $M \models_w F, F \in ATOMS$, iff $w \in P(F)$
2. $M \models_w \Diamond F$, iff $w \in W$ exists, s.th. $M \models_w F$
3. $M \models_w \Box F$, iff for all $w \in W$: $M \models_w F$
4. $M \models_w F_1 \wedge F_2$, iff $M \models_w F_1$ and $M \models_w F_2$
5. $M \models_w \neg F$, iff $M \not\models_w F$.
6. $M \models F$, iff for all $w \in W : M \models_w F$.

If $W = \emptyset$, we define $M \models F$ to be true for all formulas F (ex falso quod libet).

Lemma 3.1 *Axioms for $\mathcal{L}_{Cond}$*
The semantics of $\mathcal{L}_{Cond}$ satisfies the usual $S5$-Axioms.

When the agent changes his working hypotheses from one strategy set A to a strategy set A' this induces a change from one $\mathcal{L}_{Cond}$–model M_A to another $M_{A'}$.



So we need two concepts: A *Strategy Graph* defines an accessibility relation on the strategy sets. A connected graph of $\mathcal{L}_{Cond}$–models, the $\mathcal{L}_{Strat}$*–Model*, provides interpretation for a strategy graph.

Definition 3.6 *Strategy Graph, $\mathcal{L}_{Strat}$–Model*
A *Strategy Graph* is a structure

$$\mathcal{G} = \langle \mathcal{A}, \mathcal{R}\rangle$$

where $\mathcal{A}$ a set of strategy sets with $\emptyset \in \mathcal{A}$ and $\mathcal{R}$ is an accessibility relation on $\mathcal{A}$, i.e. $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$.

An $\mathcal{L}_{Strat}$–*Model* is a structure

$$\mathcal{M} = \langle \mathcal{W}, \mathcal{R} \rangle$$

where $\mathcal{W}$ is a set of $\mathcal{L}_{Cond}$–Models and $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$ is an accessibility relation on the $\mathcal{L}_{Cond}$–Models.

The obvious connection between the strategy graph and the $\mathcal{L}_{Strat}$–model is given by the following definition:

Definition 3.7 *Induced $\mathcal{L}_{Strat}$–Model*
Let $\mathcal{G} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a strategy graph. The $\mathcal{L}_{Strat}$–Model induced by $\mathcal{G}$ is defined as

$$\mathcal{M}_{\mathcal{G}} = \langle \mathcal{W}, \mathcal{R}' \rangle$$

where $\mathcal{W}$ is the set of $\mathcal{L}_{Cond}$–Models corresponding to the strategy sets in $\mathcal{A}$, i.e. $\mathcal{W} = \{M_A \mid A \in \mathcal{A}\}$. $\mathcal{R}'$ connects the $\mathcal{L}_{Cond}$–Models corresponding to strategy sets connected in $\mathcal{R}$: $\mathcal{R}' = \{(M_A, M_B) \mid (A, B) \in \mathcal{R}\}$.

Strategy formulas are evaluated as follows:

Definition 3.8 *Semantics for $\mathcal{L}_{Strat}$*
Let $\mathcal{M} = \langle \mathcal{W}, \mathcal{R} \rangle$ be an $\mathcal{L}_{Strat}$-Model, and S a strategy formula:

$$\mathcal{M} \models S, \text{ iff for all } M_i \in \mathcal{W} : (\mathcal{M}, M_i) \models S$$

where

1. $(\mathcal{M}, M_i) \models C, C \in \mathcal{L}_{Cond}$, iff $M_i \models C$.
2. $(\mathcal{M}, M_i) \models F_1 \wedge F_2$, iff $M_i \models F_1$ and $M_i \models F_2$.
3. $(\mathcal{M}, M_i) \models \neg F$, iff $M_i \not\models F$
4. $(\mathcal{M}, M_i) \models \Diamond F$, iff $M_j \in \mathcal{W}$ exists, such that $(M_i, M_j) \in \mathcal{R}$ and $(\mathcal{M}, M_j) \models F$

F, F_1, F_2 are strategy formulas.

This semantics is extended to a set of strategy formulas as follows: Let $\mathcal{S}$ be a set of strategy formulas:

$$\mathcal{M} \models \mathcal{S}, \text{ iff for all } S \in \mathcal{S} : \mathcal{M} \models S$$

The $\Diamond$–operator means that a working hypotheses has to be assumed in at least one of the directly following strategy sets. So in our semantics strategy formulas only propose certain working hypotheses, an agent following our semantics can adopt an arbitrary subset of these hypotheses (provided they are not contradictory together with SD and OBS). A strategy graph characterizes the solution space for strategy evaluation. A strategy is then a particular solution in this space. Until now we have declaratively defined the basic semantics for strategy formulas. In the next section we will characterize an important subclass within this semantics.

3.4 Valid Strategies

In the remainder of this chapter we will introduce the concept of a Valid Strategy which can handle the criteria we postulated in the beginning of the last section. We do this by enforcing additional restrictions on the concept of strategy graph:

1. We assume, that at least one path of the strategy graph has to end in a consistent node, where all proposed working hypotheses have already been adopted. This is captured by the concept of a *Stable Strategy Graph*.
2. We consider *monotonic* strategy graphs, i.e. each successive node only adds some working hypotheses to the strategy set of its predecessor.
3. We call a monotonic strategy graph *locally minimal*, iff only $\Diamond$–Literals necessary to satisfy the strategy formulas are added to a successor node.

Definition 3.9 *Restrictions on the Strategy Graph*
Let $\mathcal{S}$ be a set of strategy formulas, $\mathcal{G} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a strategy graph, and $\mathcal{M}_{\mathcal{G}}$ the induced $\mathcal{L}_{Strat}$–model, such that $\mathcal{M}_{\mathcal{G}} \models \mathcal{S}$.

Stable Strategy Graph $A \in \mathcal{A}$ is called a *Stable Node*, iff

it is logically consistent (M_A is not empty, i.e. there exists a set of diagnoses under strategy set A) and

$$(\mathcal{M}_{\mathcal{G}}, M_A) \models \Diamond\Box X \leftrightarrow (\mathcal{M}_{\mathcal{G}}, M_A) \models \Box X$$

$\mathcal{G}$ is called a *Stable Strategy Graph*, iff it contains a stable node $A \in \mathcal{A}$.

Monotonic Strategy Graph $\mathcal{G}$ is called monotonic, iff for all nodes $A \in \mathcal{A}$ and for each predecessor A' of A

$$A' \subseteq A$$

Locally Minimal Strategy Graph A node $A \in \mathcal{A}$ is called locally minimal, if there is no strategy graph $\mathcal{G}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ with $\mathcal{M}_{\mathcal{G}} \models \mathcal{S}$ and $A \in \mathcal{A}'$, such that

$$\{X \mid X \in WHYP \wedge (\mathcal{M}_{G'}, M_A) \models \Diamond\Box X\}$$
$$\subset \quad \{X \mid X \in WHYP \wedge (\mathcal{M}_{\mathcal{G}}, M_A) \models \Diamond\Box X\}$$

$\mathcal{G}$ is called a *Locally Minimal Strategy Graph* iff every node in $\mathcal{G}$ is locally minimal.

This means, that on each transition from A to one of its successors A' only $\Diamond$–Literals necessary to satisfy the strategy formulas are added.

The restrictions presented are summarized by the following concept:

Definition 3.10 *Valid Strategy Graph*
Let $\mathcal{S}$ be a set of strategy formulas, $\mathcal{G} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a strategy graph, and $\mathcal{M}_{\mathcal{G}}$ the induced $\mathcal{L}_{Strat}$–model, such that $\mathcal{M}_{\mathcal{G}} \models \mathcal{S}$. $\mathcal{G}$ is called a *Valid Strategy Graph* wrt S, iff $\mathcal{G}$ is stable, monotonic and locally minimal.

Definition 3.11 *Valid Strategy*
Let $\mathcal{S} \subseteq \mathcal{L}_{Strat}$. Let $\mathcal{G}$ be a valid strategy graph wrt. $\mathcal{S}$. A *Valid Strategy* is a path $\langle \emptyset, A_0, \ldots A_n \rangle$ in $\mathcal{G}$ where A_n is a stable node.

Now let us check this concept against the criteria we defined in the beginning of this section: (C1) is satisfied: if a stable strategy set exists in $\mathcal{G}$, the diagnosis agent non–deterministically chooses a path through the strategy graph ending in a stable strategy set. If no stable strategy set exists in G, G is not a valid strategy graph.

(C2) is satisfied, because every valid strategy ends in a stable node. A stable node is defined to already contain all the necessary working hypotheses. The concept of a valid strategy is a tradeoff between (C3) and efficient implementation, because it is a good approximation to minimality and we think it can still be implemented efficiently. This property of a valid strategy is made explicit in the following theorem:

Theorem 1 *Characterization of Valid Strategies*
Let $\langle \emptyset, A_0, \ldots A_n \rangle$ be a valid strategy. Let A_i, A_{i+1} be a transition between strategy sets occurring in the strategy. Then for a working hypotheses wh:

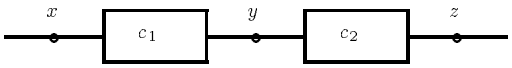
$$wh \in A_{i+1}, \text{iff}$$

wh is needed to satisfy a particular strategy formula with respect to a strategy set $A_j, j \leq i$.

Proof: At each step (going from A_j to A_{j+1}) only working hypotheses are added which are needed to satisfy the strategy formulas (local minimality). A working hypothesis remains in the strategy sets once it has been adopted (monotonicity).

3.5 An example

Consider the following system composed of two abstract components c_1 and c_2.



The abstract component c_2 has three subcomponents c_{21}, c_{22} and c_{23} not visible in the initial model.

Strategy Formulas for this example

In this small example we only consider two strategies: Structural Refinement and Behavioural Refinement. For both abstract components we have the strategy "Behavioural Refinement", i.e. we can activate a more detailed behavioural model for a component, if different diagnoses contain different fault models. Suppose, each of the components has two fault models $fm1$ and $fm2$. This can be expressed by the following formulas:

$$\Diamond mode(fm_1, c_1) \wedge \Diamond mode(fm_2, c_1) \rightarrow \Diamond\Box r\_fm(c_1)$$
$$\Diamond mode(fm_1, c_2) \wedge \Diamond mode(fm_2, c_2) \rightarrow \Diamond\Box r\_fm(c_2)$$

We use a more detailed model of c_2, if it is known to be abnormal:

$$\Box ab(c_2) \rightarrow \Diamond\Box r(c_2)$$

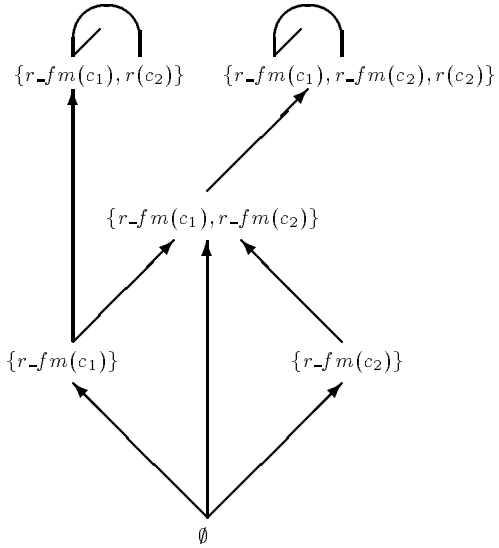
Normally we would also specify a measurement strategy for y. For simplicity, we don't provide this. Please refer to the following section.

Evaluating the strategies

Suppose, we have observed values for x and z, which are not compatible to the assumption, that both c_1 and c_2 are ok. The following table shows the diagnoses under each set of working hypotheses (only the behavioural modes are listed):

$\emptyset$	$\{mode(fm_1, c_1)\}, \{mode(fm_2, c_1)\}$ $\{mode(fm_1, c_2)\}, \{mode(fm_2, c_2)\}$
$\{r\_fm(c_1)\}$	$\{mode(fm_1, c_2)\}, \{mode(fm_2, c_2)\}$
$\{r\_fm(c_2)\}$	$\{mode(fm_1, c_1)\}, \{mode(fm_2, c_2)\}$ $\{mode(fm_1, c_2)\}$
$\{r\_fm(c_1),$ $r\_fm(c_2)\}$	$\{mode(fm_1, c_2)\}$
$\{r\_fm(c_1),$ $r(c_2)\}$	$\{mode(fm_1, c_{21})\}$
$\{r\_fm(c_1),$ $r\_fm(c_2),$ $r(c_2)\}$	$\{mode(fm_1, c_{21})\}$

Behavioural Refinement of component c_1 leads to the knowledge that c_2 must be faulty. So structural refinement has to be activated for c_2. After activating both working hypotheses the agent finds the only single fault c_{21}. The following picture shows a valid strategy graph for this example.



From the graph we can derive that $\langle \emptyset, \{r\_fm(c_1)\}, \{r\_fm(c_1), r(c_2)\} \rangle$ is a valid strategy. Under the strategy formulas we provided, also $\langle \emptyset, \{r\_fm(c_2)\}, \{r\_fm(c_1), r\_fm(c_2)\}, \{r\_fm(c_1), r\_fm(c_2), r(c_2)\} \rangle$ is a valid strategy, since $r\_fm(c_2)$ is also a possible working hypothesis under belief $\emptyset$.

3.6 Some Additional Strategies

In this section we present two additional strategies based on [1, 2] (though interpreted differently).

Introducing Physical Negation

If for a component c a behavioural mode other than the unknown mode can be assigned, we do not want to consider the unknown mode (i.e. we assume the specified fault modes are complete).

$$\Diamond mode(m_1, c) \vee \ldots \Diamond mode(m_n, c) \rightarrow \diamondsuit\Box fm\_complete(c)$$

If *fm–complete* is active for a component c, we can assure that c is assigned a known fault mode by adding the following rule to the system description:

$$\forall C : fm\_complete(C) \rightarrow (ok(C) \vee \exists M : mode(M, C))$$

Measurements

If different consistent models of the system predict different values for some component c we can discriminate between these values by making a measurement. The strategy rule used in a digital circuit would look like:

$$\Diamond val(c, 1) \wedge \Diamond val(c, 0) \rightarrow \diamondsuit\Box measure(c)$$

Sometimes this is not sufficient. When measurements are expensive, we only want to do the measurement, if the cheaper strategy st_1 is not available:

$$\Diamond val(c, 1) \wedge \Diamond val(c, 0) \wedge \neg\diamondsuit\Box st_1 \rightarrow \diamondsuit\Box measure(c)$$

means that no measurement has to be executed while the strategy literal a is still supported.

4 Summary and Future Work

The issues addressed in this paper have been first discussed in [10] and further investigated in [1, 2]. Our paper defines the concept of preferred diagnosis by a flexible and expressive preference relation between single diagnoses based on diagnosis properties. It defines the concept of diagnosis strategies using modal logic strategy formulas interpreted on a connected graph of $S5$–models. Our approach allows not only to express system models in a declarative way (which is one of the main advantages of model-based diagnosis), but extends this declarativity to the meta level by allowing the declarative description of preferences and diagnosis strategies. The integration of both concepts as presented (strategy evaluation based on our concept of preferred diagnosis) thus is a declarative definition of the results of *diagnosis as a process* which has been missing in previous approaches.

The declarative semantics for preferences and strategies can be described by extended logic programs [4]. We are currently working to efficiently implement the formal concepts introduced in this paper as well as define additional strategy classes based on this framework, (i.e. strategy classes that interpret minimality in a stronger sense).

5 Acknowledgements

We thank Carlos Damásio and Luís Pereira for their many fruitful discussions within the INIDA project. Especially Definition 2.4 and the use of modal logics for strategy formulas was jointly developed in this cooperation [4]. We especially thank Carlos Damásio for his many remarks on several versions of this paper.

References

[1] C. Böttcher and O. Dressler. Diagnosis process dynamics: Holding the diagnostic trackhound in leash. In R. Bajcsy, editor, *Proceedings of the 13th international joint conference on artificial intelligence*, volume 2, pages 1460–1471. Morgan Kaufmann Publishers, Inc., 1993.

[2] C. Böttcher and O. Dressler. A framework for controlling model-based diagnosis systems with multiple actions. *Annals of Mathematics and Artificial Intelligence, special Issue on Model-based Diagnosis*, 11(1–4), 1994.

[3] B. F. Chellas. *Modal Logic – An Introduction.* Cambridge University Press, 1980.

[4] C. V. Damásio, W. Nejdl, L. Pereira, and M. Schroeder. A declarative representation of preferences and strategies in model–based diagnosis via logic meta–programs. submitted for publication, 1994.

[5] C. V. Damásio, L. M. Pereira, and W. Nejdl. Revise: An extended logic programming system for revising knowledge bases. In *KR94*, pages 607–618, Bonn, Germany, May 1994. Morgan Kaufmann Publishers, Inc.

[6] O. Dressler and P. Struss. Back to defaults: Characterizing and computing diagnoses as coherent assumption sets. In *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI)*, pages 719–723, 1992.

[7] G. Friedrich. Theory diagnosis: A concise characterization of faulty systems. In *To appear in Proc. IJCAI 93*, Chambery, 1993.

[8] W. Hamscher, L. Console, and J. de Kleer. *Readings in Model-Based Diagnosis.* Morgan Kaufmann, 1992.

[9] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[10] P. Struss. Diagnosis as a process. In *in [8]*, 1992.

[11] P. Struss. What's in SD? towards a theory of modeling in diagnosis. In *[8]*, 1992.