Appeared in: Milne, R. (ed.): Qualitative and Model Based Reasoning for Complex Systems and their Control, Workshop KRR-4 at the 16th International Joint Conference on Artificial Intelligence, Stockholm, 1999, pp. 84-90

Model-based Support for Water Treatment

Peter Struss[1,2] and Ulrich Heller[1]

[1]Model-based Systems and Qualitative Reasoning Group of the Technical University of Munich,
Orleansstr. 34, 81667 München, Germany
[2]OCC'M Software GmbH, Gleißentalstr. 22, 82041 Deisenhofen, Germany
struss@in.tum.de, heller@in.tum.de

Abstract

Often, it is argued that there are technical processes and plants that are not amenable to model-based diagnostic or other systems because there exist no proper models, and, frequently, water treatment plants are given as a typical example. It is true that there are no models represented in classical mathematical formalisms and that component-oriented modeling, which forms the basis for classical model-based diagnosis in AI, is inappropriate for addressing the specific requirements of diagnosis and monitoring of such plants. We argue that, with a broader concept of modeling, useful formal models can be derived, especially through process-oriented modeling. This enables the exploitation of model-based techniques for diagnosis and monitoring, but this does also require significant extensions to the underlying theories and techniques: It becomes evident that they are (implicitly) tailored for diagnosing artifacts based on component-oriented modeling.

We have developed a revision of the traditional theories of diagnosis from first principles. The goal is to make it more general in terms of the class of problems to be addressed and more specific by proposing and exploiting a refined representation of the system description.

1 Introduction: Towards a Theory of Diagnosis from First Principles

Model-based diagnosis and, more specifically, consistency-based diagnosis has matured to a point where commercial tools and significant industrial applications appear. Despite this significant progress, a broad range of diagnostic tasks and potential application domains have not been addressed with consistency-based diagnosis methods equally successful, so far. Examples are diagnosis in process industry and diagnosis of disturbances in ecological systems. We argue that this is due to implicit assumptions and a very specific view underlying up-to-date consistency-based diagnosis and the limitations resulting from them. This view can be characterized in a nutshell as follows:

- The entities relevant to diagnosis are **components**, which can be associated with different behavior modes: the correct one and at least one **fault mode** (possibly with unspecified behavior).
- A system to be diagnosed consists of a **given set of** such **components** which interact in a way determined by the **fixed structure** of the system (its blueprint) and are to be scrutinized for faulty behavior.
- The result of the diagnosis is **an assignment of actual behavior modes** to all these components.
- The criterion for a proper diagnosis candidate is that the respective mode assignment **is consistent with the observations** of the actual system behavior.

The following example (see Fig. 1) is drawn from a collaborative project with DMAE, the municipal department for water and sewage of the city of Porto Alegre, Brazil. It is extremely simplified in the sense that it focuses on a particular situation and decision concerning water treatment. However, it is typical and based on a real problem encountered. Moreover, despite its simplicity, it raises a number of problems whose solution is beyond the current state of the art in model-based diagnosis.

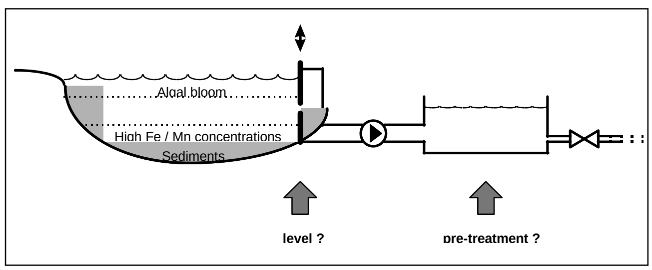

Figure 1: Water treatment at the reservoir of Lomba

Besides the huge Rio Guaíba, the reservoir of Lomba is a source for the drinking water supply of Porto Alegre. From its agricultural environment, it receives a nutrition-rich inflow that frequently causes the phenomenon of algal bloom (a burst of growth of some algae species) enabled by high temperatures in summer. Of course, water with a high concentration of algal biomass should be avoided as a source for drinking water. The intake of water from the reservoir can be adjusted in order to receive water from different levels of the reservoir. As the algae need light for their growth, the algae concentration decreases with increasing depth. which suggests a reduction of the intake level. However, the bottom water layer tends to contain a higher concentration of metals, in particular iron and manganese, which affect the color and odor of the water and whose admissible concentrations in drinking water are limited to 1mg/l and 0.5mg/l, respectively. To what extent the concentration of algae and metals varies with the water level depends, among other factors, on the turbulence of the water, which in turn is influenced by wind and, particularly, by the stratification of the water body, which can also occur in summer. "Stratification" describes the generation of different water layers with rather distinct temperature. The steep temperature gradient strongly reduces exchange processes between them. In our scenario, this phenomenon tends to confine algae to the surface layer and metals to the bottom. Taking water from the intermediate layer may, hence, be an appropriate decision. Nevertheless, the metal concentration has to be monitored and, if a threshold is exceeded, certain steps have to be taken to reduce the concentration, for instance by a pre-treatment of the water with ozone.

The ultimate task to be performed in this scenario is to determine proper actions in order to cope with the conditions in the Lomba reservoir and produce good quality drinking water. At the level discussed here, this boils down to taking two decisions (besides performing some measurements):

- which water level should be used for the intake, and
- whether or not the pre-treatment procedure has to be applied.

The foundation for these decisions is "diagnosis", i. e., for instance, starting from observations of bad water quality during or after the treatment process, determine "what goes wrong" in the current situation. But determining "what goes wrong" is quite different from standard consistency-based diagnosis which searches a device model for broken components. Although components such as pumps, valves, and pipes are involved in water treatment, it is not their malfunctioning that is causing the problem. Also it is not the case that any of the chemical, mechanical, or biological processes involved do not perform well. If one really wants to talk about a "bug" in the water treatment, it is not due to one of its existing "components", but it lies in some **missing** "component", the pre-treatment.

But this is not missing in an absolute, pre-defined structure in the process, it becomes relevant only because "something is wrong" upstream (even in the narrow sense), namely the occurrence of algal bloom in the reservoir. Obviously, "diagnosis" has to be extended to include the natural processes that affect the input to water treatment. And again, this diagnosis is not about identifying a defect component or process in this natural water body, it rather requires to find an **additional** process, namely algal bloom, that was not anticipated in the initial model of the situation. While all this distinguishes our diagnosis task from standard component-centered consistency-based diagnosis, there is another difference which concerns foundations of the theory and the starting point of diagnostic algorithms: the detection of inconsistencies. In the traditional approach, they arise from observations contradicting the system model:

$$\text{MODEL} \cup \text{OBS} \models \perp$$

Our diagnostic problem reveals that this is a specialized and narrow perspective: if our model includes the reservoir with algal bloom and high metal concentrations and the usual water treatment processes (without pre-treatment), the observation of bad drinking water quality is perfectly consistent with this model, it is just **unwanted**. While the original theory implicitly assumed that the system model carries the "gold standard" ("if all components work properly, the overall behavior is the desired one"), we now realize: what is crucial is the inconsistency of the observations with the **intended function or goals**, i. e. something that is external to the model. This is not only obvious for the processes in nature (algal bloom may be undesirable, but it is a natural process, nothing "faulty"), it also applies to technical processes.

We summarize in what respect the "classical" theories and systems of consistency-based diagnosis are too narrow and, as a result, fail to provide a solution to many diagnostic problems in the environmental domain, but also in technical applications (e. g. process industries):

- We could not call (the occurrence of) algal blooms a fault. Natural processes do not break or fail like components. This means: The relevant constituents of the system **do not have fault modes**.
- It is not the case that one of the constituents of our original system description can be blamed for the inconsistency with the observations. The reason is some **additional, unanticipated** constituent, namely algal blooms, we were not aware of. This means: a revision of the system **description cannot be confined to a set of given constituents** (the "components").
- Also, for finding an appropriate treatment, changing the “mode” (or existence) of the given constituents (e. g. by replacement of a broken component) is not the issue. One has to find actions that, again**, expand the entire system** (introduce pre-treatment).
- There are no "failures of nature". Algal blooms are not a fault, even though we might want to avoid them. A certain phenomenon may be perfectly consistent with the observations, while inconsistencies arise only **with our goals** and intentions. This means: diagnosis based on inconsistencies between the model and the observations is not the proper perspective.

As a consequence, a theory of diagnosis from first principles has yet to be developed. This paper attempts to contribute to this goal by proposing a revision and extension to consistency-based diagnosis that preserves the principled approach while expanding the scope of the underlying modeling paradigms and diagnosis tasks and algorithms

In the following section, we re-state the formal foundation of consistency-based diagnosis and its implementation. Section 3 outlines a general formalism for composing system descriptions which accommodates, in particular, component-oriented and process-oriented modeling. On this basis, the following section defines and distinguishes different tasks, including a more general notion of the “classical” diagnosis task. In the following two sections, we describe one way to implement the compositional modeling scheme and how to perform diagnosis in a manner that overcomes the limitations of the accepted wisdom of consistency-based diagnosis.

2 Consistency-based Diagnosis

The accepted wisdom of consistency-based starts from a point where a set of observations, OBS, is inconsistent with a system description, SD, and an assignment of correct behavior modes to all components (the set COMPS):

$$SD \cup OBS \cup \{OK(C_i) \mid C_i \in COMPS\} \vdash \perp \quad . \quad (1)$$

The diagnosis procedure is then organized as a search for revised mode assignments to the components that eliminate inconsistency (see e.g. [Reiter, 1987]):

$$SD \cup OBS \cup \{mode_i(C_i) \mid C_i \in COMPS\} \nvdash \perp . \quad (2)$$

Implementations such as GDE (modeling correct behavior only, [de Kleer and Williams, 1987]) and GDE+ (exploiting also fault models, [Struss and Dressler, 1989]) employ an Assumption-based Truth-Maintenance System (ATMS) in order to record inferential dependencies and to determine (minimal) sets of mode assumptions that conflict with the observations and compute diagnosis candidates as (minimal) revisions of the assignment of correct behavior to all components.

Obviously, any attempt to overcome the limitations shown above requires a more general and flexible scheme for providing the system description, SD, which is what we outline first.

3 What's in SD?

We follow the principles of structure-to-behavior reasoning and compositional modeling and provide a generalization of both component-based and process-based modeling paradigms. According to this view, the system description, SD, consists of four parts: the domain theory, the system specification (structure and parameters), the situation specification[1] and basic laws. The diagram in Figure 2 provides an overview, and we briefly discuss each part. Section 5 provides a more formal and algorithmic view.

DOMAIN THEORY
  STRUCTURE
    OBJECT TYPES
    OBJECT RELATIONS
  BEHAVIOR
    QUANTITY ASSOCIATIONS
    BEHAVIOR CONSTITUENT TYPES
SYSTEM SPECIFICATION
  (objects, relations, parameters)
SITUATION SPECIFICATION (*variables*)
BASIC LAWS

Figure 2: The Structure of SD

[1] We use the term "situation" rather than "state" in order to avoid confusion with the concept an (internal) state, e. g. in control theory. A "situation" is characterized by all variables (including state variables).

Domain Theory

The domain theory captures what we know about the domain, i. e. all systems of a certain class (e. g. hydrological ecosystems or water treatment plants). We distinguish *structure constituents* ("objects") from *behavior constituents* (which might be processes or other model fragments). The structural ontology consists of

- *object types* which occur in structural descriptions, for instance types of components in a device (resistor, broken wire), spatially distinguished entities (layers of a water body, pipes, tanks), etc. Object types can be structured hierarchically.
- *object relations* for characterizing "configurations" of objects. Examples are spatial relationships (contained-in, below), connectivity of components, etc. Some properties of relations (like uniqueness) can be specified.

The domain theory also has to provide a vocabulary for behavior descriptions and the inferences that derive behavioral constituents from a structural description. It introduces

- *quantity associations.* Parameters and state variables will be associated with instances of object types, e. g. resistance, voltage drop, and current for a resistor and manganese concentration for a water layer. For simplicity of presentation, we associate quantities only with single object instances.
- *behavior constituent types*. These are physical phenomena which are considered to contribute to the behavior of the overall system. They can represent basic component laws (Ohm's Law, logical-or) or processes in QPT ([Forbus, 1984]), such as chemical stratification, or partial behavior models like in [Bredeweg, 1991]. They occur deterministically under certain conditions, and their occurrence generates particular effects. Applying the distinction between structural aspects and the characterization of behavior through quantities to both conditions and effects, we obtain

  $$\text{STRUCT-CONDS} \wedge \text{QUANT-CONDS} \Rightarrow$$
  $$\text{STRUCT-EFFECTS} \wedge \text{QUANT-EFFECTS}$$

  as the abstract form of a behavior constituent type. Here, STRUCT-CONDS are assertions about the existence and relations of objects (e. g. existence of aenaerobic bacteria) and QUANT-CONDS are statements about values of quantities (e. g. a minimum temperature). STRUCT-EFFECTS can specify the creation and elimination of objects and their relations (e. g. the generation of some substance), and QUANT-EFFECTS can be expressed as restrictions on variables (e.g. the increase in algal biomass in dependence of temperature and nutrient concentration).

  More precisely, we state that for each constellation of objects satisfying the structural and quantity conditions, an instance of the behavior constituent is occurring - and imposes the respective effects on the constellation. An important point is that the quantity effects can be partially specified w. r. t. the combination with other behavior constituents. We use a formalism of "influences" for this purpose.

At this point, we make almost no commitment w. r. t. the quantity domains (symbolic, qualitative, real, ...), the formalism for specifying the quantity effects (constraints, differential equations, ...), and the expressiveness of structural conditions and effects (e. g. non-existence of certain objects as condition or destruction of objects as a structural effect). In general, conditions and effects are assumed to be local and compositional. Some requirements for model formation and prediction will be discussed in section 5.

System Specification

A particular system under consideration is characterized by its *object structure*, i. e. instances of the object types and individual tuples of object relations (for instance the components and the connection structure of a device) and *parameter values* for objects involved in the physical system.

Situation Specification

A particular situation of the system is characterized by *variable values*. Dependent on the task and context they may represent actual measurements (e.g. an increased amount of manganese in the drinking water), specification of goals (a certain amount of manganese), mere hypotheses, etc.

Basic Laws

Additionally, we include a part for the fundamental laws that determine the mechanisms of model formation, how influences combine and prediction over time (continuity, integration etc.). They cannot be specified arbitrarily by the modeler - but rather represent the logical equivalent of procedural aspects of model composition and prediction software components.

This way of modeling, (like QPT which can be regarded as a specialization of it), allows for dynamic changes in the set of active processes and, thus, distinguishes from approaches that represent systems as a pre-defined sequence of processes which are then considered to possibly fail very much like components (see e.g. [Guckenbiehl et al. 99]). Unlike the QPT-based work of [Collins 93], our approach includes structural effects and, hence, facilitates diagnosis w.r.t. changes in the (object-related) structure (e.g. due to unanticipated objects or interactions).

4 Characterizing Different Tasks

With the structure of the system description as a background, we can characterize problem solving tasks concerning the system. We do so using the core of consistency-based diagnosis: *Given some initial description of a system that is inconsistent with information external to the model (e.g. observations), determine a revision of this model such that consistency is achieved.*

To find out what to revise, we distinguish what we *know* about the domain, the system under consideration and its present situation, and what we *assume* about the system and the state. In particular we make the assumption explicit that we have modeled all relevant aspects of the system at hand - the closed-world assumption. In other words, this initial system description, SD_{init}, can be split into the fixed part, SD_{fix}, and the revisable part, SD_{rev}.

$$SD_{init} = SD_{fix} \cup SD_{rev}$$

where

$$SD_{rev} = STRUCT_{rev} \cup PAR\text{-}SPEC_{rev} \cup VAR\text{-}SPEC_{rev}$$

From this initial description, the domain theory and the basic laws generate a behavior model as part of some complete system description, SD_0, possibly under extension of the structure specification and completion of the parameter and variable specification. SD_0 can be inconsistent in itself or with some expectations or goals.

We can try to ask and answer different questions and characterize the task by what is considered fixed and revisable, respectively (and by specifying what establishes inconsistencies) :

- **What's going on, anyway? System identification**[2] takes observations for granted:
  $$OBS \subset VAR\text{-}SPEC \subset SD_{fix}$$
  and attempts to find appropriate system models. If model composition yields an inconsistency, one needs to find a consistent one
  $$SD_0 \vdash \perp \quad \longrightarrow \quad SD_1 \nvdash \perp$$
  by revising some of the initial assumptions about the structure and parameter values:
  $$SD_{rev} = STRUCT_{rev} \cup PAR\text{-}SPEC_{rev}, \quad \text{or just}$$
  $$SD_{rev} = PAR\text{-}SPEC_{rev}$$
  for **parameter identification** for a system with known structure.
- **What's going on right now? Situation assessment** treats the system as fixed and tries to determine variable values starting from observations and (possibly revised) assumptions about values:
  $$OBS \cup STRUCT \cup PAR\text{-}SPEC \subset SD_{fix}$$
  This may boil down to prediction of values, but can also include revision of assumptions about the situation and doubtful observations:
  $$SD_{rev} = VAR\text{-}SPEC_{rev}$$
- **What's going wrong? Diagnosis:** What is "wrong", is determined by some goal, rather than physics itself. This means inconsistencies and the necessity for diagnosis are caused by criteria that are external to the physical system and its model (e. g. a limit on the concentration of manganese). We have to make these goals explicit. If they are violated by the system description (possibly SD_1 obtained in the previous step - if we assume it represents the actual system sufficiently well) one can ask how SD_1 has to be revised in order to no longer contradict the goals:
  $$SD_1 \cup GOALS \vdash \perp \longrightarrow SD_2 \cup GOALS \nvdash \perp.$$
  The "difference" between SD_1 and SD_2, i. e. the revised elements of SD_1, which eliminate the contradiction can be considered to have "caused" the trouble. Diagnosis may focus on ultimate or external causes such as unexpected objects being present or try to identify values that

[2] Precisely what is happening is **model identification**. But note that, in contrast to system identification in engineering, the model generated is not merely a mathematical model, but stated in terms of physical phenomena!

can be influenced for treatment or control. We assume that GOALS are stated as a characterization of healthy or desirable situations, i. e. as variable specifications:

$\text{GOALS} \cup \text{OBS} \subset \text{VAR-SPEC} \subset \text{SD}_{fix}$

$\text{SD}_{rev} = \text{STRUCT}_{rev} \cup \text{PAR-SPEC}_{rev}$

- **What can be done? Control and therapy proposal** has to look for revisions that establish consistency with the goals and can be achieved by real *actions*:

  $\text{SD}_1 \cup \text{ACTIONS} \cup \text{GOALS} \not\vdash \perp$,

  or, stronger, to establish them:

  $\text{SD}_1 \cup \text{ACTIONS} \vdash \text{GOALS}$.

  In some cases, the actions correspond to the revisions of the diagnosis task, e. g. removing a disturbing object, replacing an identified faulty component by a correct one, manipulating a deviating parameter etc. This could be considered as a "real cure", which eliminates the ultimate causes of a disturbance. Then a (complete) diagnosis uniquely determines the therapy. This might be impossible and the proposal of structural changes by adding objects and object relations and/or modify parameters and variables might be the only control option. This corresponds to real "therapy" and is the interesting case considered in this paper.

  Introducing pre-treatment with ozone, which might not be prepared in the initial plant model, but is rather selected from the domain theory as a possible action, illustrates the case

  $\text{SD}_{rev} = \text{STRUCT}_{rev}$

  whereas reconfiguration of the water distribution network by changing the states of valves means

  $\text{SD}_{rev} = \text{VAR-SPEC}_{rev}$.

  These examples also highlight that the possible means for therapy influence or even determine the formulation of the diagnosis task and that the framework includes the possibility of a "therapy without diagnosis". However, elaboration of this is beyond the scope of this paper.

  If some GOALS are defeasible, they may be considered for revision, as well.

5 SD in Action: Model Formation, Prediction and Revision

The reasoning tasks discussed above include important non-monotonic steps. This is a consequence of the usage of the closed-world assumption to calculate quantities, which may in turn have effects on the occurrence of other behavior constituents. This makes implementations difficult and often inefficient. However, for certain domain theories, we have been able to implement a model composition algorithm that employs monotonic reasoning and some form of constraint satisfaction.

5.1 Instantiation and Activity

In the following, we consider the case, where we have only positive structural effects, i. e. objects (and relations) are only generated, but not destroyed, and only positive structural conditions (i. e. no conditions in terms of non-existence of objects or relations). If we assume, furthermore, that we can determine the identity of any generated object (e. g. using some relation like spatial locatedness by "contained-in"), then we can monotonically construct all objects that can *possibly* be generated by any instance of a behavior constituent, without taking the quantity values into account.

This corresponds to the common distinction between "instantiation" and "activation", as described in the Qualitative Process Theory ([Forbus, 1984]), which does not allow for structural effects in the domain theory. Technically, we separate the structural conditions and structural effects from the quantity conditions and quantity effects.

In an instantiation phase, also called *model composition* (see Figure 3), all potentially active behavior constituents are assembled, i. e. for each set of objects matching the structural conditions of a behavior constituent type, we create an instance of a behavior constituent, bc_i, plus all objects and relations created in the associated structural effects.

Some guard variables are introduced along with constraints that block structural effects from becoming effective before the respective quantity conditions are met. Namely, for each object instance and each relation tuple, we introduce an additional boolean quantity exists, representing whether it is present or not. A behavior constituent does have a similar boolean variable active_{bci}, encoding whether both its structural and quantity conditions are met. A set of constraints encodes the following dependencies:

$\text{STRUCT-CONDS} \wedge \text{QUANT-CONDS} \Leftrightarrow \text{active}_{bci}\text{=true}$

$\text{active}_{bci}\text{ =true} \Rightarrow \text{STRUCT-EFFECTS}$

Where STRUCT-CONDS and STRUCT-EFFECTS are expressed using the exist quantities of the respective objects and relations.

In the second phase (constraint generation in Figure 3), the behavior constituent instances are turned into a behavior model that can be used for prediction. For quantity effects, locally specified as constraints,

$\text{constraint}(x_1, x_2, ..., x_n)$

(where x_j represent quantities) we add a condition, so that their effects can be switched on and off according to the current value of the active variable. The resulting constraint

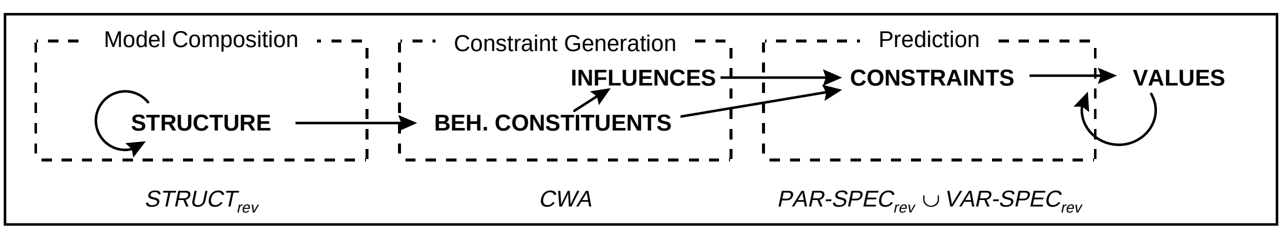

Figure 3: Phases and Assumptions for Model Formation and Prediction

is

$\text{active}_{bci}\text{=true} \Rightarrow \text{constraint}(x_1, x_2, ..., x_n)$

For partially specified quantity effects, i. e. influences, we carry out influence resolution. Here, we have to take into account that the influences are only potentially active, and we further explicitly assume that the collected influences are the only ones acting on the influenced variable (local closed-world assumption). For a set of influences from quantities x_1, ... x_n on variable y, this is achieved by duplicating the x_j yielding x_1', ... x_n', with additional constraints

$\text{active}_{bcji} = \text{true} \Rightarrow x_i' = x_i,\ \forall i$

$\text{active}_{bcji} = \text{false} \Rightarrow x_i' = 0,\ \forall i$

$y = \text{sum}(x_1', x_2', ..., x_n')$

where active_{bcji} are the activity variables of the behavior constituent instances creating the influences from x_i to y. The constraints are written for the (standard) case of additive influence combination, so that a neutral contribution is generated for inactive influences. The third one is the resolution constraint, which is supported by the local closed-world assumption for y, CWA_y. With this step, the influences have been combined and converted into sets of constraints.

As a third phase, the system performs *prediction* in its general sense, using the known and assumed quantity values. This means evaluating the quantity effects plus the newly introduced constraints for controlling the effects: behavior constituents can become "active" or "inactive", which triggers or suspends both their structural and quantity effects.

This way, we can compute a representation of the extensions of the non-monotonic theory (maximum consistent sets of consequences, see [Reiter, 1980]) by prediction. A constraint propagator can create (most of) the intersection of all extensions, while a complete constraint filtering algorithm could even generate all extensions.

When the identity of newly created objects cannot be determined, we face unification problems with existing objects. So we have to construct multiple worlds - and can rely solely on the predictions that are identical in all worlds.

Further note, that the approach can fail entirely, if model formation produces infinitely many potentially active behavior constituents, even if only a finite number could become active under the given quantity specification.

5.2 Assumption Tracking

Our main objective in prediction is the detection of conflicts, i.e. sets of assumptions that are inconsistent with the system description, observations, and goals. For this purpose, we have to keep track of the

- structural assumptions ($STRUCT_{rev}$) leading to some quantity effect being present in the model at all (or, rather, being active at a particular moment),
- of the closed-world assumptions used in combining influences (CWA) and all
- quantity value assignments that are considered revisable ($PAR\text{-}SPEC_{rev} \cup VAR\text{-}SPEC_{rev}$).

Thus, we have requirements for assumption tracking in all three phases (see Figure 3).

5.3 Searching for Revisions

For performing one of the tasks defined in section 4, the system starts off with the constraint network generated during model formation. Observations or goals are added in the form of variable specifications. Hence, consistency checking is a matter of further prediction only, and conflicts, i. e. inconsistent assumption sets can be generated as in component-oriented diagnosis. The candidates generated from these conflicts (also as usual) include assumptions to be revised. They fall in one of three categories: variable assignments (elements of $PAR\text{-}SPEC_{rev} \cup VAR\text{-}SPEC_{rev}$), existence of objects or relation tuples (elements of $STRUCT_{rev}$), and closed-world assumptions (See Figure 4).

The structural and quantity assumptions can immediately be tested for consistency by prediction with the revised system description (all effects of the objects or relation tuples now revised to be non-existent are constructed with conditional constraints, so that they will be switched off consistently as by the constraints described in section 5.1).

Closed-world assumptions are of a different nature, for their retraction does not give a clue what the revised model should look like. It is only the starting point for a search. We are looking for a revision (in terms of objects, relation tuples and quantity values) that violates the local closed-world assumption by hypothesizing behavior constituents whose influences could remove inconsistencies. This revision will be a candidate in meaningful, i. e. ontological, terms.

The domain theory provides the necessary background knowledge to determine the behavior constituent types, which could influence a quantity of the given kind (associated object-type) at all. Each behavior constituent instance in question would require to have both its structural and quantity conditions satisfied. Each possibility can be subject to further search, e. g. the structural conditions of one behavior constituent instance could be established by the structural effects of another one with even simpler conditions.

Of course, in carrying out the search within the space of the domain theory, we rely on the domain theory to be complete w. r. t. to the phenomena of interest.

Collins [1993] is solving the analogous problem for QPT

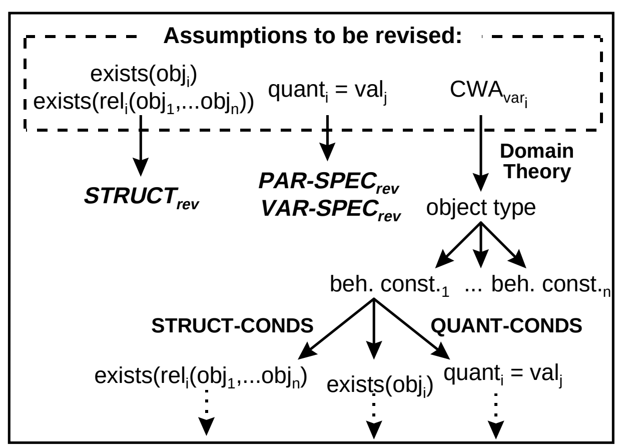

Figure 4: Searching for Revisions

models. He proposes an analogous distribution of the global closed-world assumption and is using an abductive back-chaining algorithm to generate explanations implying the violation of the local closed-world assumption. But since he is not considering structural effects of processes, structural revisions are taken as primitives.

Generally, focusing the search is an important problem to be solved, since standard minimality principles can be misleading. Even in the initial candidate generation, where it is common to prefer the candidates with the minimal number of elements, it is possible that a larger candidate with a number of (violations of) local closed-world assumptions can be explained by a smaller revision, e. g. by introducing a single object triggering a range of additional processes achieving the deviating behavior.

This problem recurs with every intermediate search result, so that in general we have no guarantee that there is no simpler candidate. One solution is to distinguish a class of revisions as "ultimate causes".

The search triggered by hypothesized violations of the closed-world assumptions provides significantly more power than traditional, component-oriented diagnosis, but, obviously, has its price in terms of complexity. One should notice, however, that there is a class of diagnostic cases that require the introduction of additional objects, but not necessarily the retraction of the closed-world assumption and the costly search: if the respective object has been created by the model formation step, but the system description implies its non-existence (exists = false), due to the given specification of quantities, this is directly reflected by "classical" candidates that are subsets of VAR-SPEC_{rev}.

6 Perspectives and Discussion

The described approach to different tasks of diagnostic problem solving attempts to preserve the formal and rigorous foundation, while overcoming the specific conditions and limitations of state-of-the-art consistency-based diagnosis. It should help to expand the scope of applicability significantly.

Classical component-oriented diagnosis is a (very) special instance of this scheme. However, our extension offers a principled approach to diagnosis of structural faults. Furthermore, it allows for appropriate diagnostics in cases where component replacement is not a means for repair, but we rather aim at reconfiguring the system. We also feel that extended consistency-based diagnosis can complement fault diagnosis and identification (FDI) methods developed in control engineering by providing the search for model revisions at the level of physical phenomena rather than mathematical expressions.

We have implemented model formation as described and plan to use a commercial constraint-based diagnosis tool that provides an efficient ATMS, a predictor and a candidate generator. We will use this as the core of an integrated decision support system for environmental applications, but also potentially for applications in the process industry.

Among the unsolved problems is the handling of unification problems in object creation (if two objects of the same type are created, they could be distinct or identical, which spans two "worlds"). It is not yet clear, how predictions in one world could be shared in another.

Some issues of guiding and focusing the search for revisions when retracting a closed-world assumption have been pointed out. However, it is not easy to derive useful heuristics for the general case. Solutions could come from exploiting certain structures of specific domain theories.

References

[Bredeweg, 1991] Bredeweg, B., Expertise in Qualitative Prediction of Behaviour. Doctoral thesis, Faculty of Psychology, University of Amsterdam, 1991.

[Collins, 1993] Collins, J. W., Process-based Diagnosis: An Approach to Understanding Novel Failures. Doctoral thesis, Institute for the Learning Sciences, Northwestern University, 1993.

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

[Forbus, 1984] Forbus, K., Qualitative Process Theory. Artificial Intelligence, 24(1-3), 1984.

[Guckenbiehl et al. 99] Guckenbiehl, T., Milde, H., Neumann, B., Struss, P., Meeting Re-use Requirements of Real-life Diagnosis Applications. Puppe, F. (ed.), XPS99: Knowledge-based Systems. Lecture Notes in Artificial Intelligence 1570, Springer Verlag, Berlin 1999, pp. 90-100

[Hamscher et al., 1992] Hamscher, W., Console, L., and de Kleer, J., Readings in Model-based Diagnosis. Morgan Kaufmann, San Mateo, 1992.

[Reiter, 1980] Reiter, R., A logic for default reasoning. Artificial Intelligence, 13:81-132, 1980.

[Reiter, 1987] Reiter, R., A Theory of Diagnosis from First Principles. Artificial Intelligence, 32(1):57-96, 1987.

[Struss and Dressler, 1989] Struss, P., Dressler, O., Physical Negation: Integrating fault models into the general diagnostic engine. International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, 1989.