The Institute For Research In Cognitive Science

Goal-Directed Diagnosis - Diagnostic Reasoning in Exploratory-Corrective Domains

by

Ron Rymon

University of Pennsylvania
Philadelphia, PA 19104-6228

October 1993

Site of the NSF Science and Technology Center for Research in Cognitive Science

P E N N

University of Pennsylvania
Founded by Benjamin Franklin in 1740

IRCS Report 93-36

Goal-Directed Diagnosis – Diagnostic Reasoning in Exploratory-Corrective Domains*

Ron Rymon
Department of Computer and Information Science
University of Pennsylvania
Room 419C, 3401 Walnut Street, Philadelphia, PA 19104
E-mail: rymon@linc.cis.upenn.edu

(Proceedings IJCAI-93)

Abstract

In many diagnosis-and-repair domains, diagnostic reasoning cannot be abstracted from repair actions, nor from actions necessary to obtain diagnostic information. In general, in *exploratory-corrective* domains an agent has to interleave exploratory activity with activity aimed at achieving its goals. In TraumAID 2.0, a consultation system for multiple trauma management, we implement a reasoning framework for such domains which integrates diagnostic reasoning with planning and action. This paper presents *Goal-Directed Diagnosis* (GDD), a formalization of TraumAID 2.0's diagnostic reasoning. Taking the view that a diagnosis is only worthwhile to the extent that it can affect repair decisions, GDD uses *goals* to focus on such. Goals are also useful as a means of communicating with its accompanying planner.

1 Background and Motivation

In many domains, it is common to distinguish reasoning and activity concerned with *what* problems need be addressed from that reasoning concerned with *how* to address those problems. As such, Artificial Intelligence (AI) subsumes as separate sub-disciplines *diagnosis* research, seeking the source (or sources) of a system's faulty behavior, and *planning* research, concerned with the construction of action plans to achieve certain goals. Based on that dichotomy, most formalizations of diagnosis aim at a *diagnosis object* as a solution.

In some domains, however, this may be inadequate. In trauma management, for one, *therapy* is the ultimate objective and diagnosis is the "price" that one has to pay in order to achieve that objective. In such domains, we argue, diagnosis should only persist so long as it can affect those decisions for which it was carried out in the first place, namely repair decisions. We call this the *Goal-Directed Diagnosis* (GDD) *principle*.

The GDD principle suggests that diagnostic and therapeutic decisions be considered together. Furthermore, in many diagnosis-and-repair domains multiple diagnostic and therapeutic needs require, and compete for, the agent's *activity*. In [Rymon, 93], we propose an *Exploratory-Corrective Management* (ECM) architecture (Figure 1) employing a basic cycle of diagnostic reasoning, planning and action.

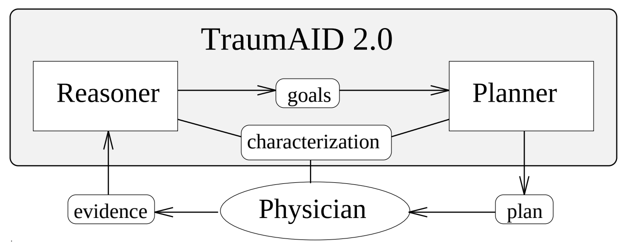

Figure 1: The ECM Architecture

In this architecture, diagnostic and therapeutic goals are uniformly posted to the planner, who is charged with addressing them *as a combination*. For exploratory-corrective domains such as trauma management, the ECM architecture satisfies the following desiderata:

1. it allows interleaving diagnosis and repair.
2. it positions the diagnostic reasoner to
   (a) set diagnostic and therapeutic goals;
   (b) use incoming evidence to monitor actions and other events, reason about changes in knowledge or state, and adapt goals;
3. it positions the planner to mediate between concurrent diagnostic and therapeutic needs.

TraumAID is a consultation system for the diagnosis and treatment of multiple trauma (for an overview see [Webber *et al.*, 92]). TraumAID 2.0, its new ECM-based version, has recently been judged by a panel of trauma surgeons as *significantly more acceptable than actual care.*[1]. This paper formalizes TraumAID 2.0's *goal-directed* diagnostic reasoning. It is structured as follows: Section 2 reviews related work; Section 3 formalizes GDD; and Section 4 discusses its use within the ECM framework to implement a variety of diagnosis-and-repair *strategies*.

*This work was supported in part by a graduate fellowship, ARO Grant DAAL03-89-C0031PRI, and by a National Library of Medicine grant 1RO LM051217-01.

[1] Three trauma surgeons were asked to blindly compare the actual care in 97 trauma cases to the management that would have been proposed by TraumAID 2.0. The judges indicated a significant preference for TraumAID 2.0 plans over actual plans by a ratio of 64 to 17 with 16 ties ($p < 0.001$ by binomial test). For more details see [Rymon, 93].

2 Related Work

2.1 Formalizations of diagnosis

Recent years have seen significant advances in formal approaches to diagnosis. A large number of approaches and frameworks have been suggested: probabilistic classifiers and discriminators, logical consistency-based and abductive paradigms, graph-based formulations in which causal and coincidental relations are modeled, etc. What is common, however, to all these formalizations is that they take a *diagnosis object* – broadly defined as a characterization of the current state of affairs – as a solution and therefore as their goal. Our Goal-Directed Diagnostic paradigm takes a different view of diagnosis and its objectives: *recommendations are important!*.

To understand this view, consider first that diagnosis is rarely an independent task, but rather serves the purpose of another process, e.g. repair. With incomplete information, general theories of diagnosis will often give rise to a large number of hypothetical diagnoses. As it often happens, many of these are irrelevant. Their computation, and the need on the part of the matrix process to sort through them, are thus a waste.

An important observation made by Poole and Provan [90] is that the optimality of a diagnosis must depend on *post-diagnosis* goals. To that end, they advocate the use of utilities [91b]; in [91a] they further note that there is often *no need* for a complete explanation and that the granularity of a solution again depends on its uses (and also on available tests).

In some domains, explicitly specified utilities can be used to reflect issues of importance to the matrix process in diagnostic process decisions. However, while utility-theory sees increased use in Artificial Intelligence in general and medical decision making in particular, it also requires a level of completeness and precision in characterizing a domain that is sometimes hard to obtain and may not be available. The GDD principle can be viewed as a qualitative analogue of a utility-maximization principle. The GDD framework supports the implementation of this principle in ECM agents by explicitly representing and reasoning about goals. In particular, alternative actions are being ruled in or out according to direct and indirect (i.e. through action) goal interactions, e.g. suppression, subsumption, compatibility, preferences, etc.

In exploratory-corrective domains, it is often hard to separate diagnosis from repair. The GDD paradigm is thus part of a *total* approach for reasoning which combines diagnostic reasoning, planning and action. Friedrich *et al.* [91] and Sun and Weld [92] share much of this view. Considering diagnosis as part of an overall diagnosis-and-repair process, Friedrich *et al.* note that repair does not always require a complete diagnostic explanation. Unlike other formalizations, their theory has no explicit notion of a diagnosis object. Instead, a sequence of tests and repair actions is sought, that if applied to the current state, will imply (as in a logical proof) a restoration of the diagnosed system to a proper working condition. Presented not as a theory of diagnosis but as a theory of *repair planning*, their work applies a possible-models planning approach [Winslett, 88] to a diagnostic domain. Friedrich and Nejdl [92] describe a set of algorithms for diagnosis-and-repair plans. Sun and Weld use UWL, a STRIPS-like language, in an approach which integrates GDE-style diagnosis and STRIPS-style planning. The link between diagnosis and repair planning in *real* applications is also emphasized by Pepper and Kahn [87]. Also related, although to a lesser extent, is work by Rushby and Crow [91] who formalize reconfiguration, a form of repair, using an extension of Reiter's [87] theory of diagnosis. In GDD, we use goals to focus on repair-worthy issues. In the goal-level, we use GDD rules to resolve *direct* interactions between goals; *indirect* interaction between goals is resolved by the accompanying planner.

In exploratory-corrective domains, actions are often necessary to obtain diagnostic information. While this is also true of many other domains, sequential diagnostic frameworks often take a simplistic view of information acquisition. Often what is considered is the potential, or expected, discriminatory power of a given piece of information. The potential ramifications of diagnostic activity, even on the very condition it is aimed at diagnosing, are often *not* considered; most models assume costless questions, or at best attach a simple cost to each piece of information. However, rather than worrying about these issues, research in diagnosis can simply rely on planning research which, studying these issues extensively, should come handy. Goals, the *architectural duty* of the GDD reasoner, serve as a natural interface with an accompanying planner.

2.2 AMORD

AMORD [deKleer *et al.*, 77] is a general purpose problem solver which is accompanied with truth-maintenance and planning facilities. The main thesis behind AMORD's reasoning component is that combinatorial forward-chaining can be avoided via meta-reasoning, i.e. if the problem solver reasons explicitly about its reasoning strategy. In particular, AMORD's reasoner posts *inference goals*, distinct from those posted to its planner, which are used to *control* reasoning. GDD shares this intuition. The key to the differences between AMORD's reasoner and GDD's are the distinct objectives of their matrix systems: AMORD's objective is to control reasoning, whereas in the ECM architecture our purpose is to control actions. Thus, the goals posted by GDD serve none of its own purposes but are rather aimed at the planner. Although some of GDD's goals are aimed at knowledge, knowledge goals are only encoded when *action* is required. In the ECM architecture, GDD is used to explicitly encode *local* strategies, and goal-level interaction between strategies. Most of the mediation and coordination between a number of concurrently pursued strategies is done by the planner *implicitly*, i.e. on the fly, as the diagnosis-and-repair session proceeds, based on general principles [Rymon, 93].

3 Goal-Directed Diagnostic Framework

Goal-Directed Diagnosis (GDD) begins with the point of view that diagnosis is only worthwhile if it has the potential to affect future decisions. Thus, while we accept the common definition of a *diagnosis* as a case characterization, we believe that *different purposes* can lead to *different characterizations* of the same situation. For example, different purposes may lead to different refinement efforts. GDD allows explicit encoding of purposes, which it uses to guide its problem solving. More specifically, throughout a problem solving session, the GDD reasoner will maintain both a *belief* – a description of the current characterization, and an *attitude*

– encoding a sense of purpose by pointing to goals worth pursuing.

In a recurrent cycle, the GDD reasoner takes as input a diagnostic problem, characterized by (1) observations; and (2) mappings (rules) from from observations to conclusions (belief), and from observations and conclusions to goals (attitude). A solution to such problem is a new attitude-belief assignment. Goals, propositions regarded as relevant by the current attitude, are the addressed by the accompanying planner and served by the actor (the physician in our case). New observations result in a modified diagnostic problem, and a new cycle is initiated.

In this section, we describe a rule-based language for specifying diagnostic problems in GDD a corresponding inference scheme.

3.1 Underlying framework: Multi-valued logics

Multi-Valued Logics (MVL) [Ginsberg, 88] is a formal framework for inference in which each proposition is assigned not only a truth value, corresponding to the strength of belief in that proposition being *true or false*, but also a knowledge assessment, measuring roughly the amount of knowledge used to derive such belief. Bilattices, in which one partial order ($\leq_t$) corresponds to the truthfulness measure and the other ($\leq_k$) to the knowledge one, are used in MVL as domains for truth-value assignment. Bilattice values are then combined along the truthfulness dimension using the regular $\vee$ and $\wedge$ operators (join and meet of the $\leq_t$ lattice). The $+$ operator (join in the $\leq_k$ lattice) can be used to combine knowledge sources. Figure 2 depicts the smallest non-trivial bilattice, with four points: **T**rue, **F**alse, **U**nknown, and $\perp$ (representing the presence of contradictory evidence).

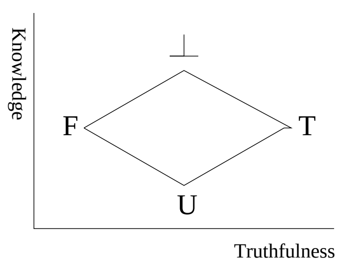

Figure 2: Basic Truth-Knowledge Bilattice

We have first formalized GDD using a three-valued logic: *true*, *false* and *unknown* [Rymon *et al.*, 91]. The MVL reformulation buys us expressive flexibility (e.g. by extending the domain bilattices with default values). To fit our needs, we have specialized MVL to a rule-based representation; Ginsberg's original formulation uses a full first-order logic, and thus requires an underlying theorem prover. While the material presented here is self-contained, the reader is referred to [Ginsberg, 88] for a more complete coverage of MVL.

3.2 Attitude and belief

During the diagnostic process, the GDD reasoner will maintain and update an *attitude* and a *belief* for propositional statements. To remain general, propositions may be any fact about the patient or the world that the reasoner may know to hold, may know not to hold, may assume, may want to know whether hold, may want to achieve, may be confused about, etc. The reasoner's attitude towards and/or belief in a given proposition will change over time as a result of new information becoming available, new inferences drawn, activity carried out, etc.

In GDD, each proposition is assigned a value drawn from the cross product of *two* (possibly distinct) bilattices: one representing belief, the other attitude (or desire). The notion of belief is interpreted regularly, whereas the attitude component represents problem-solving control information and measures the relevance of acquiring information about, or achieving the condition described by, the particular proposition. The belief bilattice still has the truthfulness and knowledge partial orders; the attitude bilattice has relevance ($\leq_r$) and knowledge dimensions.

Definition 3.1 *Attitude and Belief*

Given a set of primitive propositions $\mathrm{H} \stackrel{\mathrm{def}}{=} \{h_i\}_{i=1}^n$, an *attitude* maps H to an attitude bilattice B_A; a *belief* maps H to a belief bilattice B_B. An *attitude-belief* combines the two mapping H to $B_A \times B_B$. Conversely, an attitude-belief can also be viewed as a pair $\langle \phi_A, \phi_B \rangle$ of attitude and belief functions.

Although not necessarily so, for the purpose of this paper we shall assume that both B_A and B_B are 4-point bilattices. The belief bilattice, following Ginsberg's suggestion, is defined by the truth-knowledge partial orders. In the attitude bilattice, the truth partial order is replaced with a *relevance* measure. Notably, one's knowledge with respect to the truthfulness of a proposition need not equal, in general, one's knowledge with respect to the relevance of that same proposition. Technically, our attitude bilattice is made of **R**elevant, **I**rrelevant, **U**nknown, and $\perp$ values. Extensions to more complex bilattices are discussed in [Ginsberg, 88; Rymon, 93]

3.3 Goals

Goalhood is a *semantic* interpretation of an attitude-belief assignment. Generally speaking, a proposition p is a goal if its *attitude* assignment is highly relevant (in our bilattice if $\phi_A(p)$=R). Of course, not every relevant proposition is an *operational* goal. If $\phi_B(p)$=T, for example, then we believe p holds and may want to consider it "achieved". (in more complex bilattices, one has to define which combinations of relevance and achievement levels need be addressed, in what order of preference, etc.)

The GDD inference, we should note, is purely syntactic and is thus indifferent to these semantic subtleties; within the ECM architecture, it is the planner's role to decide which goals to pursue. However, we wish to note here a potential distinction that is based on whether a proposition denotes a diagnostic (knowledge) goal or a therapeutic (state achieving) one, namely that while a diagnostic goal may often be regarded satisfied whenever the proposition is proved *either* true or false, we may want to actually achieve (i.e. $\phi_B(p)$=T) a therapeutic one.

3.4 Representation

We use rules to represent knowledge. Two types of rules are used: one for inferring belief, the other for inferring attitude (goals). Antecedents in both types of rules are stated in belief *terms*. A rule's consequent (header) must be either a proposition p or its negation $\neg p$.

Definition 3.2 *Rules*

1. *Evidential rules* are used to infer belief. For example, the following rule concludes whether a patient's shock is due to abdominal bleeding:

   Shock $\wedge$
   $\neg$*Single_wound_to_upper_chest* $\wedge$
   unless(Pericardial_Tamponade) $\wedge$
   unless(Massive_Hemothorax) $\wedge$
   unless(Tension_Pneumothorax)
   $\Rightarrow$*Shock_of_possible_abdominal_origin*
2. *Goal Setting rules* are used to infer attitude. For example, the following rule concludes whether it is relevant to know whether a patient has hematuria:

   Gunshot_wound_to_abdomen $\wedge$
   Bullet_in_abdomen
   $\triangleright$ *Hematuria*

A proposition p can head a *number* of goal-setting *and* evidential rules. In particular, a goal-setting rule headed by p reflects that it is worth to pursue knowledge about p, or the state described by p, depending on whether p is semantically used as a diagnostic or repair goal. An evidential rule headed by p is used to conclude whether or not it holds, or put differently whether or not it has been satisfied.

Example 3.3 Consider the diagnosis and repair of a *pericardial tamponade*[2]. Throughout that process, the following diagnostic and therapeutic goals are instantiated, addressed, and satisfied:

1. **Setting a Diagnostic (knowledge) goal:**
   $(\ldots)$ $\triangleright$ *Pericardial_Tamponade*
   "It is necessary to *know* if the problem exists".
2. **Satisfying a Diagnostic (knowledge) goal:**
   $(\ldots)\Rightarrow$ *Pericardial_Tamponade*
   "Conclude that the problem exists".
3. **Setting a Therapeutic goal:**
   $(\ldots)$ $\triangleright$ *Relieve_pressure_pericardial_sac*
   "It is necessary to *address* the problem".
4. **Satisfying a Therapeutic Goal:**
   $(\ldots)\Rightarrow$ *Relieve_pressure_pericardial_sac*
   "The problem has been successfully addressed".

While rules used by GDD to express knowledge have their antecedents expressed solely in belief terms, it may often be useful to predicate goalhood of one proposition on the relevance (or irrelevance) of another goal. To facilitate this within a belief-based antecedent calculus, we added a mapping (*attitude*) from the attitude bilattice to the belief bilattice, roughly modeling the belief in the relevance of a given proposition.

Definition 3.4 *A Diagnostic Problem*
A *diagnostic problem* is a quadruple $P \stackrel{\text{def}}{=} \langle$H,RB,M$_0$,OBS$\rangle$ such that:

- H=$\{h_1, h_2, \ldots, h_n\}$ is a set of propositions;
- RB is a set of evidential and goal-setting rules;
- $M_0 \subseteq$H is a set of observations;
- OBS : $M_0 \rightarrow B_B$, is a partial belief function.

[2]A condition in which blood fills the pericardial sac, interfering with the heart's operation.

3.5 Solving a diagnostic problem

Solving a diagnostic problem requires computing the inferential closure of the observations, given the rules. Our closure definition is based on Ginsberg's, and can be viewed as its simplification to the rule-based case.

Definition 3.5 *Inferential Closure*
An attitude-belief $\langle \phi_A, \phi_B \rangle$ is an *inferential closure* for a problem instance P iff

1. It coincides with OBS, i.e $\forall$ h$\in M_0$ ϕ_B(h)=OBS(h);
2. For any proposition $d \in$H–M$_0$, let $\{R_i\}_{i=1}^k$ be all the evidential rules with d in their header, $\{\overline{R}_i\}_{i=1}^l$ all those with $\neg d$ in their header, then
$$\phi_B(d) = \sum_{i=1}^{k}(\phi_B^*(body(R_i)) \vee U) + \sum_{i=1}^{l}(\neg\phi_B^*(body(\overline{R}_i)) \wedge U);$$
   Where $\phi_B^*(body(R))$ represents the belief term obtained from the conjunction of antecedents of the rule R.
3. Similarly, for any proposition $d \in$H, let $\{R_i\}_{i=1}^k$ be all the goal-setting rules with d in their header, $\{\overline{R}_i\}_{i=1}^l$ all those with $\neg d$ in their header, then
$$\phi_A(d) = attitude^{-1}(\sum_{i=1}^{k}(\phi_B^*(body(R_i)) \vee U) + \sum_{i=1}^{l}(\neg\phi_B^*(body(\overline{R}_i)) \wedge U));$$

Essentially, all rules *for* a proposition and *against* it are weighted as individual *knowledge sources*, and thus combined using the $+$ operator. The correspondence between this formulation and Ginsberg's is that there *all proofs* for a given statement have to be weighted whereas here the only proofs are the specific rules. Even with this limited scope, we should note that in general there is no guarantee that such inferential closure is unique, computable, or even exists. However, the following straightforward algorithm has worked for us so far:

Algorithm 3.6 *Computing an Inferential Closure*

1. Start off with the observations, by setting
   $\phi_A(h) \stackrel{\text{def}}{=} U$, for all h$\in$H
$$\phi_B(h) \stackrel{\text{def}}{=} \begin{cases} OBS(h) & h \in M_0 \\ U & \textit{otherwise} \end{cases}$$
2. Forward-chain on the rules, enforcing conditions 2 and 3 above, until reaching a fixed point.

The formal definition of a solution in the GDD framework emphasizes its distinction from other diagnostic frameworks.

Definition 3.7 *A Diagnosis*
Let $\langle \phi_A, \phi_B \rangle$ be the inferential closure for a problem instance P, then ϕ_B is a *diagnosis* for P.

Most formalizations of diagnosis take a *diagnosis* as their solution. In GDD, however, we associate more importance with the goals (and consequently the actions) adopted during the diagnosis process. In the ECM architecture, solving the current diagnostic problem has an operational purpose: it defines the goals to be pursued next by the planner.

Definition 3.8 *A Solution to a Diagnostic Problem*
A *solution* to a diagnostic problem P is the *complete* inferential closure.

4 The Diagnosis-and-Repair Process

So far, we have described how a single diagnostic problem is defined, and how it is solved to produce a new set of goals and conclusions. In this section, we describe how GDD is *used*, in conjunction with the ECM's planner, to produce desirable management plans.

4.1 The ECM algorithm: integrating diagnostic reasoning and planning

Algorithm 4.1 calls the diagnostic reasoner whenever new evidence defines a new diagnostic problem. The solution, particularly the relevant goals, guides the complementary planner in the choice of activity which, in turn, may start a new cycle.

Algorithm 4.1 *ECM Diagnosis-and-Repair Algorithm*

1. Initialize $\langle \phi_A, \phi_B \rangle$ to coincide with OBS;
2. Compute an inferential closure for $\langle \phi_A, \phi_B \rangle$;
3. Construct a plan P for the combination of goals indicated by that closure;
4. Unless P is empty do
   - Execute P until new evidence arrives;
   - Update $\langle \phi_A, \phi_B \rangle$ to reflect this evidence;
   - Go to step 2.

Note that the termination criterion is not necessarily related to the concreteness of the working diagnosis. The process terminates when the plan is empty, i.e. when all goals have been addressed, or no means are available for addressing remaining goals, etc.

4.2 Mediating between local strategies

The decomposition of reasoning in the ECM architecture suggests a way to encode *strategies*, appropriate responses to anticipated situations, in it. We do that by using GDD to explicitly encode *local* strategies, and using planning to implicitly merge (on-line) a patient-specific *combination* of such. In particular,

- In the trauma management domain, strategies can be localized around diagnosis and/or treatment of a single problem, or of a common combination of problems. The rules abstracted in Example 3.3, for instance, can be viewed as part of a strategy for diagnosis and treatment of a pericardial tamponade: pursuing the knowledge goal *Pericardial_Tamponade* established in (1) would hopefully result in a state in which it is concluded whether or not it holds (2), and so on. Strategies can be described as sequences, or more generally as directed acyclic graphs (for instance when there are multiple alternatives to pursue a goal or when there are a number of alternative outcomes).
- Managing a given patient may require merging a number of strategies, e.g. for multiple problems. In merging local strategies, we must reason about potentially adverse interactions, and may wish to take advantage of potential synergies. In the ECM architecture, interaction between strategies is resolved in two levels: the goal level and the action level.

  Most of the interaction between strategies must be resolved in the action level. Briefly, given a combination of goals stemming from multiple strategies, the planner has to choose a combination of actions and order them respectively. In our formulation, a number of alternative procedures (action sequences) can be indicated for each goal. Each of these has its advantages and disadvantages depending on the patient's state, available resources, other problems, etc. The potential for synergy comes from the fact that often times a procedure can serve a number of goals at once. Our *selection-and-ordering* planning algorithm [Rymon, 93] uses domain considerations (prioritization principles, constraints, etc.) to construct a plan that synergetically combines the multiple strategies.

  Certain interactions can be resolved in the goal level without having to consider particular choices of action. Specifically, goal-setting rules can be used to suppress, inhibit, or prioritize strategies or parts thereof. For example, the following rule asserts that, in unstable patients, the pursuit of a tension pneumothorax has a higher priority than that of a pericardial tamponade:

  Shock $\wedge$
  relevant(Tension_Pneumothorax) $\wedge$
  $\neg$*known(Tension_Pneumothorax)* $\wedge$
  $\triangleright$ $\neg$ *Pericardial_Tamponade*

  Resolving interaction in the goal level may be advantageous in some situations given the combinatorics of planning, and disadvantageous in others given that it has to be encoded explicitly in GDD rules.

Finally, we wish to note that, for engineering purposes, abstract versions (templates) of commonly used local strategies can be stored in a library and instantiated when appropriate. In [Rymon, 93] we present some commonly used strategies, as well as strategies that mix diagnosis and repair in a variety of intriguing ways.

4.3 Example

To illustrate multiple cycles in the ECM architecture, we next follow limited aspects of a case from the initial observations and the suspected diagnosis, to its validation, treatment, and effectiveness verification.

Consider a patient presenting in the emergency room in stable condition, suffering a gunshot wound to the left chest. A new diagnostic problem is instantiated with these observations. Let $\langle \phi_A, \phi_B \rangle$ denote the system's current attitude-belief. Initially ϕ_A(h)=ϕ_B(h)=U, for all propositions h$\in$H. As soon as the observations are reported, ϕ_B is set accordingly.

Next, the closure of $\langle \phi_A, \phi_B \rangle$ is computed. In particular, we use the following goal-setting rule to set the *diagnostic goal* of knowing whether or not the patient suffers a hemothorax [3].

(1) *Chest_Wound(Left)*
$\triangleright$ *Simple_Hemothorax(Left)*

At this point, control is transferred to the planner which recommends a chest X-ray as a diagnostic means. In the

[3] A condition reflecting internal bleeding in the chest cavity.

presence of other problems, the planner will have to order the X-ray with respect to actions aimed at other needs. Different means may be selected if more efficient given the combination of goals.

Suppose the physician orders an X-ray and reports signs of hemothorax and fractured ribs. Each of these findings will then be updated in the system's belief, and may trigger further investigation. The hemothorax finding will trigger the following evidential rule:

(2) *X_ray_shows_Simple_Hemothorax(Left)*
$\Rightarrow$ *Simple_Hemothorax(Left)*

The system thus updates ϕ_B(*Simple_Hemothorax(Left)*) from U to T. That change may be interpreted as a satisfaction of the diagnostic goal set by (1). Note too that we must distinguish a hemothorax finding from the condition of having a hemothorax, since the condition can be diagnosed in other ways, e.g. through the presence of decreased breath sounds. The diagnosed hemothorax triggers the following goal-setting rule:

(3) *Simple_Hemothorax(Left)*
$\triangleright$ *Rx_Simple_Hemothorax(Left)*

The attitude toward this *therapeutic goal* is updated from U to R. It is referred to the planner which recommends addressing it through the insertion of a chest tube. Then, evidence that a chest tube has been inserted leads to another diagnostic goal of verifying its proper placement and that it is functioning. In addition to these two cycles of reasoning and activity (a subsequent X-ray is required to check proper placement), the following rule is evaluated to check that the original therapeutic goal is *actually* satisfied:

(4) $\neg$*Chest_tube_misplaced(Left)* $\wedge$
Chest_tube_is_functioning(Left) $\wedge$
Chest_tube_is_draining_blood(Left)
$\Rightarrow$ *Rx_Simple_Hemothorax(Left)*

In summary, the hemothorax condition is tracked from the initial wound report, through its investigation, diagnosis, treatment, and verification.

5 Summary

We presented a formalization of *Goal-Directed Diagnosis* (GDD), and have briefly described its use within the ECM framework. In the ECM architecture, GDD is used to post both diagnostic and therapeutic goals to an accompanying planner. The planner is used to mediate between multiple needs. Importantly, assuming that diagnosis is only worthwhile to the extent that it can affect repair decisions, GDD focuses on recommendations (goals), rather than on explanatory characterization. Goals are also convenient as a natural interface with the planner.

Acknowledgement

Many of the ideas presented were shaped while implementing TraumAID 2.0 and as a result of discussions within the TraumAID group. Particular thanks go to Bonnie Webber, John Clarke, Greg Provan and Charlie Ortiz. I would also like to thank Peter Szolovits for important comments and suggestions, Matt Ginsberg for helping me understand MVL, and anonymous reviewers.

References

[deKleer *et al.*, 77] deKleer, J., Doyle, J., Steele, G., and Sussman, G., Explicit Control of Reasoning. *MIT AI Memo No. 427*, 1977.

[Friedrich *et al.*, 91] Friedrich, G., Gottlob, G., and Nejdl W., Formalizing the Repair Process. *Proc. 2nd Int'l Workshop on Principles of Diagnosis*. pp. 11-22, Milan, Italy, 1991.

[Friedrich & Nejdl, 92] Friedrich, G., and Nejdl W., Choosing Observations and Actions in Model-Based Diagnosis/Repair Systems. *Proc. 3rd Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pp. 489-498, Cambridge MA, 1992.

[Ginsberg, 88] Ginsberg, M., Multivalued Logics: A Uniform Approach to Inference in Artificial Intelligence. *Computational Intelligence*, 4:265-316, 1988.

[Pepper & Kahn, 87] Pepper, J., and Kahn, G., Repair Strategies in a Diagnostic Expert System. *Proc. 10th Int'l Joint Conference on Artificial Intelligence*, pp. 531-534, Milano, Italy, 1987.

[Poole & Provan, 90] Poole, D. and Provan G., What is an Optimal Diagnosis? *Conference on Uncertainty in Artificial Intelligence*, pp. 46-53, 1990.

[Poole & Provan, 91a] Poole, D., and Provan G., Use and Granularity in Consistency-Based Diagnosis. *Proc. 2nd Int'l Workshop on Principles of Diagnosis*. pp. 1-10, Milan, Italy, 1991.

[Provan & Poole, 91b] Provan, G., and Poole, D., The Utility of Consistency-Based Diagnostic Techniques. *Proc. 2nd Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pp. 461-472, Cambridge MA, 1991.

[Reiter, 87] Reiter, R., A Theory of Diagnosis From First Principles. *Artificial Intelligence*, 32, 1987, pp. 57-95.

[Rushby & Crow, 91] Rushby, J. and Crow, J., Model-Based Reconfiguration: Toward an Integration with Diagnosis. *Proc. 9th National Conference on Artificial Intelligence*, pp. 836-841, Anaheim CA, 1991.

[Rymon *et al.*, 91] Rymon, R., Webber, B., and Clarke, J., Towards Goal-directed Diagnosis (Preliminary Report). *Proc. 2nd Int'l Workshop on Principles of Diagnosis*. pp. 23-39, Milan, Italy, 1991.

[Rymon, 93] Rymon, R., *Diagnostic Reasoning and Planning in Exploratory-Corrective Domains.* Ph. D. Thesis. In preparation. Department of Computer and Information Science, University of Pennsylvania, 1993.

[Sun & Weld, 92] Sun, Y., and Weld, D., Beyond Simple Observation: Planning to Diagnose. *Proc. 3rd Int'l Workshop on Principles of Diagnosis*, Seattle WA, 1992.

[Webber *et al.*, 92] Webber, B., Rymon, R. and Clarke, J., Flexible Support for Trauma Management through Goal-directed Reasoning and Planning. *Artificial Intelligence in Medicine* 4(2), pp. 145-163, 1992.

[Winslett, 88] Winslett, M., Reasoning about Action Using a Possible Models Approach. *Proc. 7th National Conference on Artificial Intelligence*, pp. 89-93, St. Paul MN, 1988.