REPRESENTATION AND USE OF EXPLICIT JUSTIFICATIONS FOR KNOWLEDGE BASE REFINEMENT

Reid G. Smith *Howard A. Winston*
Tom M. Mitchell[1] *Bruce G. Buchanan*[2]

Schlumberger-Doll Research
Old Quarry Road
Ridgefield, CT 06877-4108
USA

ABSTRACT

We discuss the representation and use of justification structures as an aid to knowledge base refinement. We show how justifications can be used by a system to generate explanations – *for its own use* – of potential causes of observed failures. We discuss specific information that is usefully included in these justifications to allow the system to isolate potential faulty supporting beliefs for its rules and to effect repairs.

This research is part of a larger effort to develop a *Learning Apprentice System* (LAS) that partially automates initial construction of a knowledge base from first-principle domain knowledge as well as knowledge base refinement during routine use. A simple implementation has been constructed that demonstrates the feasibility of building such a system.

I. INTRODUCTION

A *Learning Apprentice System* [6] is an interactive aid for building and refining a knowledge base. Its aims are twofold: *(i)* to partially automate initial construction of a knowledge base by generating shallow rules automatically from an approximate domain theory; and, *(ii)* to interact with users to help refine the knowledge through experience gained during normal problem solving. In this paper, we concentrate on knowledge base refinement. In our scenario, for problems where the Performance Program (*i.e.*, the component of the knowledge-based system that performs the problem solving) fails to make an important inference, or makes an incorrect inference, the user advises the LAS of the failure. The system tries to explain why the failure has occurred (perhaps via a focussed interaction with the user), and repairs or extends its knowledge base as required.

This type of *"knowledge acquisition in context"* has previously been used to advantage in TEIRESIAS [3, 1], in the context of the MYCIN system. Its advantages include: *(i)* it happens in the normal course of use of the system, driven by failures, and therefore has the potential of tapping the knowledge of specialists without placing large additional demands on their time; and, *(ii)* because the specialist and system are working step by step through a specific problem, the learning is focussed on a relatively small portion of the knowledge base.

Our main focus here is explanation of failures – assignment of blame to specific items in the knowledge base. In terms of the general model of learning systems presented in [2], this is the task of the Critic. Like TEIRESIAS, we use the reasoning trace exhibited by the Performance Program to help focus the interaction between the LAS and the user. However, instead of relying on the user to explain why a particular rule or set of rules might have failed, the LAS uses a type of dependency network – a *justification structure* – to construct possible explanations for the observed failure.

The reasoning trace provides support for conclusions drawn by the Performance Program. The content of that support is the chain of rule firings that led to the conclusions. A *justification structure*, on the other hand, goes one step further. It records the support *for an individual rule*. The content of the justification is the deeper domain knowledge from which the rule is derived, together with the assumptions and approximations that have been used in the derivation. For the purpose of refinement, the primary information captured in a justification structure is the manner in which errors propagate across dependency links. By using this information in concert with a taxonomy of error types, it is possible for the LAS to determine a constrained set of suspect supporting beliefs that can explain an observed failure of some supported belief. This information is useful for focussing a dialogue between the LAS and the specialist. It is also essential input to an automated revision system.

Several researchers [5, 8] have studied methods for recording logical dependencies between beliefs, and for using such dependencies to guide inference. The current work extends the concepts of truth maintenance and dependency-directed backtracking in that domain-specific knowledge about how to propagate errors through a dependency network is used in addition to knowledge of the logical structure of the network itself.

The XPLAIN system of Swartout [10] also used justifications to generate explanations of its behavior for human users. By constrast, we focus here on using justifications to generate explanations *for the system itself* of potential causes of observed failures. This has led us to concentrate on determining precisely which information is usefully included in the justifications to allow the system to isolate potential faulty supporting beliefs for its rules and to effect repairs.

We are using the *Dipmeter Advisor** system in order to test our ideas in the context of a task (and Performance Program) that is already well understood [9]. We have implemented subprograms to extend this system and have undertaken some analysis of the generality of the method.

---

[1] T. Mitchell is with the Computer Science Department, Rutgers University, New Brunswick, NJ 08903.

[2] B. Buchanan is with the Computer Science Department, Stanford University, Stanford, CA 94305.

* Mark of Schlumberger

A. GEOLOGICAL INTERPRETATION: THE DIPMETER ADVISOR SYSTEM

The general task of the *Dipmeter Advisor* system is to infer the geological structures penetrated by a single borehole. For example, it might conclude that a late fault with strike 92° has been penetrated at a depth of 1050 feet. Its primary input data is the dip or tilt of rock formations penetrated by the borehole, indexed by depth. It also uses data from a variety of other logging tools as well as information about the local geology.

The system divides the interpretation task into an ordered series of subtasks involving pattern detection, data aggregation, and abstraction. After each subtask has been completed, the user is given the opportunity to examine, delete, or modify conclusions reached by the system. He can also add his own conclusions. In addition, he can revert to earlier stages of the analysis and repeat various subtask chains with different assumptions or data.

The rules in the system are empirical associations applied by the rule interpreter in distinct rule sets using a forward-chaining control strategy. A simplified version of a rule used to determine that a previously detected normal fault is a *late* fault is shown in Figure I-1.

Rule: NFR12

IF
there exists a normal fault pattern (p), and
there exists a red pattern (p1),
such that the length of p1 < 50 ft., and
such that p1 is above the fault plane pattern of p,
THEN
specialize p to be a late fault pattern

Figure I-1: Late Fault Rule.

The geometry and dip patterns upon which NFR12 is based are shown in Figure I-2. A rough justification for the rule is as follows: A late fault typically has a distortion region directly above the fault plane. The distortion region for a late fault is generally thin because the surrounding rock has been compacted, and is therefore not very plastic, at the time of faulting. The thin distortion region, projected onto the borehole, is manifested in dipmeter data as a short red pattern. The 50 ft. threshold is dependent upon assumed values for the distortion region thickness and the fault angle. The rule assumes that dipmeter patterns can be detected in the vicinity of the fault. This will be true if bedding is preserved, the borehole is not washed out (caved in), and if the tool is operating correctly. If, for example, the borehole were washed out over part of the distortion region, the length of the measured red pattern would be less than expected.

We use this rule as an example throughout the paper. We show how its detailed justification can be used to isolate faulty supporting beliefs that can prevent it from firing in some situations.

II. DESCRIBING ERROR TYPES AND RULE JUSTIFICATIONS

The Critic uses the type of error that the Performance Program has made, along with its reasoning trace, the current rule base, and justification structures to guide generation of plausible explanations of a failure and to focus on plausible repairs.

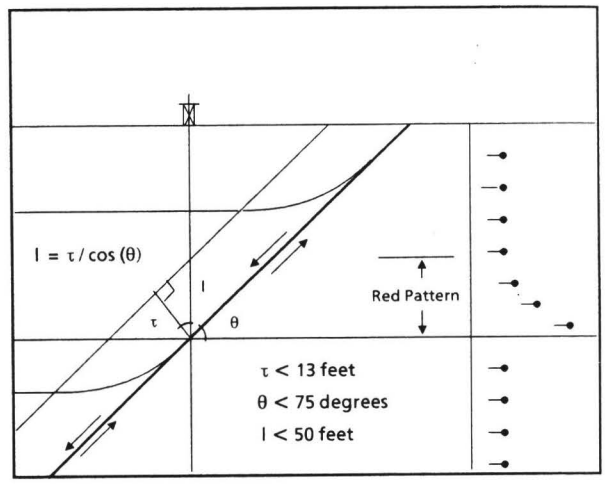

Figure I-2: Late Fault Geometry

A. ERROR TYPES

In examining the justification of an incorrect rule, it is useful for the Critic to distinguish among several classes of errors because: *(i)* it provides an additional constraint on the search for suspected supporting beliefs; and, *(ii)* different classes of errors suggest different repair strategies. The error classes currently used by our prototype LAS for beliefs are shown in Figure II-1, and break down into three main classes:

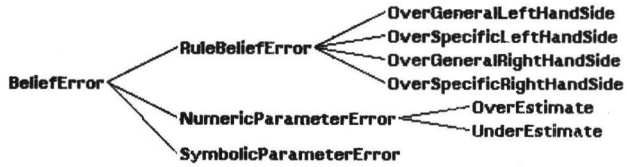

Figure II-1: Types of Belief Errors

Rule Belief Error: Beliefs that are implications may suffer from one of the following classes of error: *OverGeneralLeftHandSide* (OGLHS) – the rule applies in situations in which it should not; *OverSpecificLeftHandSide* (OSLHS) – the rule does not apply in situations in which it should; *OverGeneralRightHandSide* (OGRHS) – the rule fails to make assertions when it should or makes assertions that are too general (*e.g.*, drawing a conclusion about a normal fault where a conclusion about a late fault, a specialization of normal fault, is warranted); and, *OverSpecificRightHandSide* (OSRHS) – the rule either makes assertions when it should not or makes assertions that are too specific (*e.g.*, drawing a conclusion about a late fault where only a conclusion about a normal fault is warranted).

Numeric Parameter Error: A belief about the value of a numeric parameter may be incorrect by being an *OverEstimate* (OE) or *UnderEstimate* (UE) of the correct value of the parameter.

Symbolic Parameter Error (SPE): This is an error regarding beliefs which are neither implications nor assertions about numerical parameters. They simply have a truth value. For example, an erroneous assertion that a comparator is "<" when it should be ">" is a symbolic parameter error.

We believe this error taxonomy to be useful in a variety of domains. It is also incomplete. We have not yet formalized, for example, errors in symbolic parameters for which a partial order can be established. In such cases, *OverSpecific* and *UnderSpecific* may be appropriately added as subclasses of *SymbolicParameterError*.

B. JUSTIFICATION STRUCTURES

We define a justification structure to be a network of *Beliefs* connected by *Justification Links*. Beliefs represent assertions and Justification Links record the derivational dependencies between beliefs.

A *Belief* contains *Assertion*, *Type Of Belief*, and *Degree Of Belief* slots (in addition to its links in the justification structure). The Assertion is the actual statement of the belief. The Type Of Belief is one of *definitional* (*i.e.*, no further justification is required), *theoretical* (*i.e.*, based on the half-order theory – could be incorrect if the theory is incorrect), *statistical* (*i.e.*, justified by statistical experience), or *default* (*i.e.*, cannot be estimated without information either typically unavailable to the system or expensive to obtain). The manner in which the Type Of Belief of a justified belief is propagated from its justifier beliefs is determined from the Linkage Rule (below) that enables the justification.

The Degree Of Belief is a numerical measure of the validity of beliefs whose Type Of Belief is statistical or default. We have not yet explored its use.

A *Justification Link* contains *Linkage Rule* and *Error Propagation* slots (in addition to its links in the justification structure).

The Linkage Rule points to the rule of inference that enables the justified belief to be derived from the justifier beliefs (*e.g.*, modus ponens). It is described in more detail in the next section.

The Error Propagation specifies the way in which errors in justifier beliefs propagate to the justified belief This form of this information for a particular link is inherited from the Linkage Rule for the link. It allows the Critic to focus on particular justifier beliefs that might be responsible for an error of a specific type in the justified belief.

This slot is filled with a list of the form

```
((<error-type> <clause>
   (<error-type> <clause> <justifier>) ...) ...)
```

For example, suppose the Error Propagation slot for the Justification Link of the rule *Belief17* has the value *((OSLHS Clause1 (OSLHS Clause2 Belief16)))*. This structure states that if the rule asserted by *Belief16* suffers from an *OverSpecificLeftHandSide* error in *Clause2* of its left-hand side then *Clause1* of the left-hand side of *Belief17* could suffer from the same type of error.

C. LINKAGE RULES

A linkage rule specifies the rationale that allows a justified belief to be derived from its justifier beliefs. It can be either *Truth-Preserving* or *Non-Truth-Preserving*. The only type of truth-preserving rule is *Deductive*. It is essentially a logical proof of the validity of the derivation.

While we hypothesize a number of Non-Truth-Preserving linkage rules, the only one we have used to date is the *Abductive* rule. Abductive inference allows the conclusion of A→B to be drawn from B→A and A. For example, from the quasi-theoretical statement that meningitis causes fever and an observation of fever in a patient, abductive inference permits the *conclusion* that the patient is suffering from meningitis. Because fever is commonly associated with a number of other causes as well, the backward (interpretive) form of the causal rule can only be used to *suggest* the cause when the manifestation is observed. (For a general discussion of Abduction, see [7].)

A linkage rule also contains *Type of Belief Propagation* and *Error Propagation* slots.

The *Type of Belief Propagation* slot specifies the way that the Types of Belief of justifier beliefs are propagated to beliefs justified via the rule. Deductive rules, for example, propagate the minimum of Type Of Belief of the justifier beliefs, according to the following partial order: definitional, theoretical, statistical, default. The Abductive rule is a default-producing rule. Regardless of the Type Of Belief associated with the justifier beliefs, default is propagated as the Type Of Belief of the justified belief.

The *Error Propagation* slot specifies a template for propagating errors from justifier beliefs to justified beliefs.

D. EXAMPLE: THE NFR12 JUSTIFICATION

The hand-generated NFR12 justification structure is shown in Figure II-2. Beliefs are indicated by nodes with names B i. Justification Links are shown between beliefs, but are not explicitly named. A capsule summary of some of the beliefs is shown. We show the entire justification structure to give some feel for its complexity. The actual number of nodes should not be taken too seriously because there is considerable latitude available with respect to what constitutes a belief and what constitutes a linkage rule. We have inserted knowledge of the task domain as beliefs and knowledge that is not specific to the domain as linkage rules (*e.g.*, geometry and algebra).[3]

The general form of the justification is that it is based on the relative positions of three-dimensional regions associated with a fault. When these regions are penetrated by a borehole, measured data will be related to the zones that are the projections of the regions on the borehole, as manifested in patterns seen by a particular tool (*e.g.*, the dipmeter tool).

Nodes that are boxed in the figure are described in detail in the following. Most linkage rules are *Deductive*. We only note a linkage rule when it has some special significance (*e.g.*, if it is *Non-Truth-Preserving*). The logic encoding of each each belief assertion is shown. The reader may find it helpful to simply scan the justification at this point and refer back to it while following the example presented in Section A. We only show Error Propagation information that is relevant to the example. The complete justification for NFR12 is found in [11].

B1: If p is a Normal Fault Pattern and there exists a Red Pattern, p1, such that the length of p1 is less than 50 ft. and p1 is Above the fault plane pattern of p, then p is a Late Fault Pattern.

$$\forall p\ \exists p1\ [[NFP(p) \wedge (RP(p1) \wedge (length_p(p1) < 50) \wedge Above_p(p1, fpp(p)))] \rightarrow LFP(p)]$$

Type Of Belief: default

[3] Rule justifications provide support for the inferences made in the reasoning trace. Similarly, we might consider construction of another layer of justification – support for the inferences made in linking beliefs in the rule justification. We have not yet examined this sort of multi-layer justification.

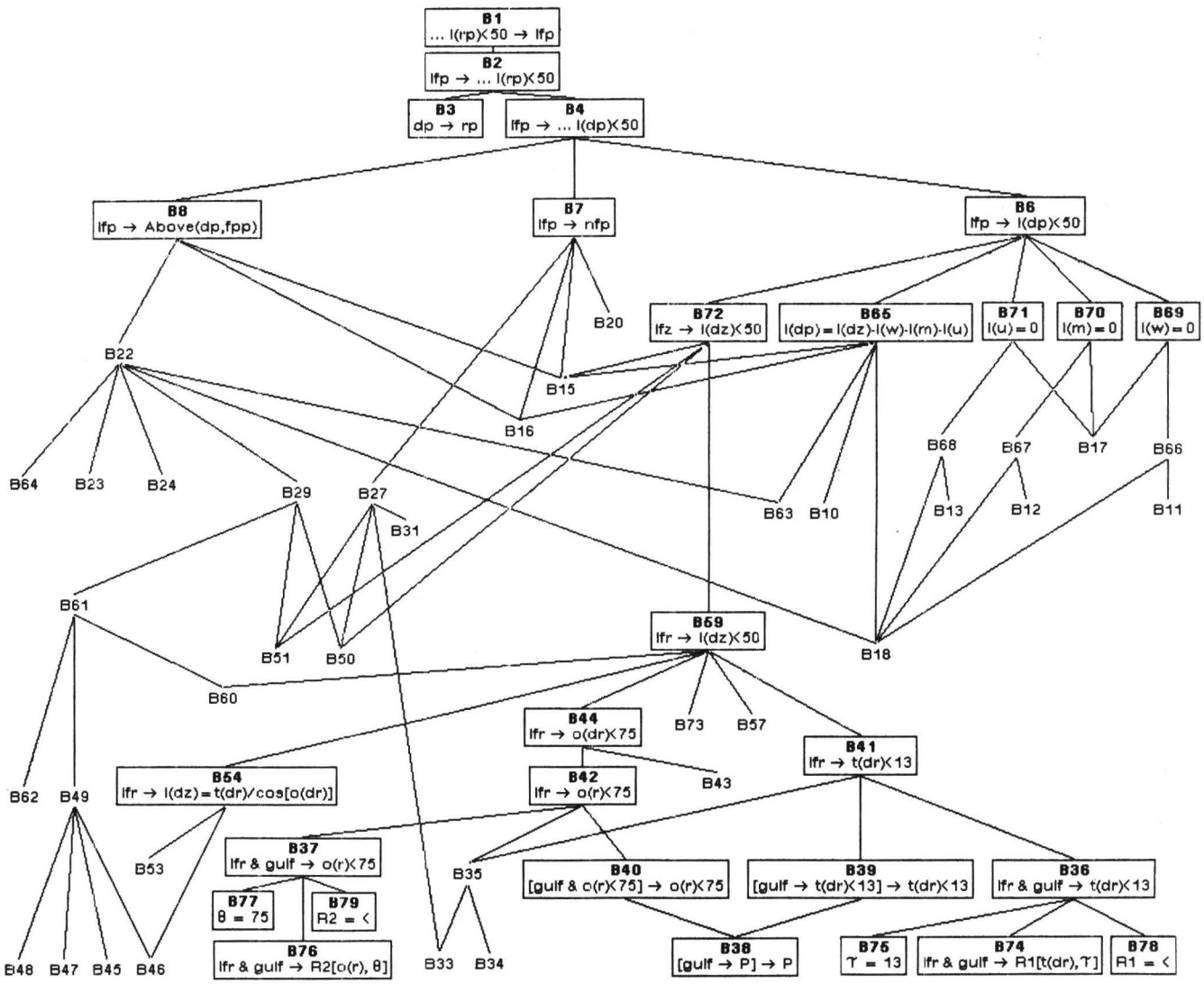

Figure II-2: Justification for NFR12

Linkage Rule: Abductive
Error Propagation: ((OSLHS C2 (OSRHS C2 B2)) ...)
Note: This belief is supported by a non-truth-preserving linkage rule.

B2: If p is a Late Fault Pattern, then p is also a Normal Fault Pattern and there exists a Red Pattern p1 such that the length of p1 is less than 50 ft. and p1 is Above the fault plane pattern of p.

$\forall p\ \exists p1\ [LFP(p) \rightarrow [NFP(p) \wedge (RP(p1) \wedge (length_p(p1) < 50) \wedge Above_p(p1, fpp(p)))]$

Type Of Belief: default
Error Propagation:
((OSRHS C2 (OSRHS C2 B4) (OSRHS C1 B3)) ...)

B3: The Distortion Pattern of a pattern is called a Red Pattern

$\forall p,q\ [DPofP(p, q) \rightarrow RP(p)]$

Type Of Belief: definitional

B4: If p is a Late Fault Pattern, then p is also a Normal Fault Pattern and there exists a p1 such that p1 is the Distortion Pattern of p, the length of p1 is less than 50 ft., and p1 is Above the fault plane pattern of p.

$\forall p\ \exists p1\ [LFP(p) \rightarrow NFP(p) \wedge [DPofP(p1, p) \wedge (length_p(p1) < 50) \wedge Above_p(p1, fpp(p))]]$

Type Of Belief: default
Error Propagation:
((OSRHS C2 (OSRHS C1 B6) (OSRHS C1 B7) (OSRHS C1 B8)) ...)

B6: The length of the distortion pattern of any Late Fault Pattern is less than 50 ft.

$\forall p\ [LFP(p) \rightarrow (length_p(dp(p)) < 50)]$

Type Of Belief: default
Error Propagation:
((OSRHS C1 (OSRHS C1 B65) (OSRHS C1 B72))
(OGRHS C1 (OSRHS C1 B69) (OSRHS C1 B70) (OSRHS C1 B71)) ...)

B7: $\forall P\,[LFP(p) \rightarrow NFP(p)]$
Type Of Belief: definitional
Note: This is supported by beliefs that relate regions to zones and zones to patterns.

B8: If p is a Late Fault Pattern, the distortion pattern of p is Above the fault plane pattern of p.
$\forall p\,[LFP(p) \rightarrow Above_p(dp(p), fpp(p))]$
Type Of Belief: theoretical

B65: If p is a Late Fault Pattern, the length of the distortion pattern of p is equal to the length of the corresponding distortion zone minus the sum of the lengths of the distortion zone's washouts, mirror images, and unpreserved bedding zones.
$\forall p\,[LFP(p) \rightarrow length_p(dp(p)) = length_z(dz(z_p(p))) - length_z(w(dz(z_p(p)))) - length_z(m(dz(z_p(p)))) - length_z(ub(dz(z_p(p))))$
Type Of Belief: theoretical

B69: If p is a Late Fault Pattern, the length of the washout zones of the distortion zone of the zone associated with p is 0.
$\forall p\,[LFP(p) \rightarrow (length_z(w(dz(z_p(p)))) = 0)]$
Type Of Belief: default

B70: If p is a Late Fault Pattern p, the length of the mirror image zones (*i.e.*, zones where the tool is not operating correctly) of the distortion zone of the zone associated with p is 0.
$\forall p\,[LFP(p) \rightarrow (length_z(m(dz(z_p(p)))) = 0)]$
Type Of Belief: default

B71: If p is a Late Fault Pattern p, the length of the unpreserved bedding zones of the distortion zone of the zone associated with p is 0.
$\forall p\,[LFP(p) \rightarrow (length_z(u(dz(z_p(p)))) = 0)]$
Type Of Belief: default

B72: If p is a Late Fault Pattern, the length of the distortion zone of the zone of p is less than 50 ft.
$\forall p\,[LFP(p) \rightarrow (length_z(dz(z_p(p))) < 50)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (OSRHS C1 B59)) ...)

B59: If r is a Late Fault Region, the length of the distortion zone of the zone of r is less than 50 ft.
$\forall r\,[LFR(r) \rightarrow (length_z(dz(z_r(r))) < 50)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (OSRHS C1 B41)
(OSRHS C1 B44)
(OSRHS C1 B54)) ...)

B41: The thickness of the distortion region of any Late Fault Region is less than 13 ft.
$\forall r\,[LFR(r) \rightarrow (t(dr(r)) < 13)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (OSRHS C1 B36)) ...)

B44: The orientation of the distortion region of every Late Fault Region is less than 75°.
$\forall r\,[LFR(r) \rightarrow (o(dr(r)) < 75)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (OSRHS C1 B42)) ...)

B42: The orientation of every Late Fault Region is less than 75°.
$\forall r\,[LFR(r) \rightarrow (o(r) < 75)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (OSRHS C1 B37)) ...)

B54: If r is a Late Fault Region, the length of the zone of the distortion region of r is equal to the thickness of the distortion region of r divided by the cosine of the orientation of the distortion region.
$\forall r\,[LFR(r) \rightarrow (length_z(z_r(dr(r))) = t(dr(r))/\cos(o(dr(r))))]$
Type Of Belief: definitional

B36: If r is a Late Fault Region in the Gulf Coast, the thickness of r is less than 13 ft.
$\forall r\,[LFR(r) \wedge G(r) \rightarrow (t(dr(r)) < 13)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (UE B75) (SPE B78)) ...)
Note: Either the threshold (13 ft.) or the comparator (<) could be incorrect.

B39: $\forall r\,[R(r) \wedge (G(r) \rightarrow (t(dr(r)) < 13))] \rightarrow (t(dr(r)) < 10)$
Type Of Belief: definitional

Note: This is an instantiation of P in **B38** to be the distortion region thickness condition.

B40: $\forall r\,[R(r) \wedge (G(r) \rightarrow (o(r) < 75))] \rightarrow (o(r) < 75)$
Type Of Belief: definitional
Note: As in **B39** with P the orientation condition.

B37: If r is a Late Fault Region in the Gulf Coast, its orientation is less than 75°.
$\forall r\,[LFR(r) \wedge G(r) \rightarrow (o(r) < 75)]$
Type Of Belief: statistical
Error Propagation: ((OSRHS C1 (UE B77) (SPE B79)) ...)
Note: Analogous to **B36.**

B38: For all regions r and predicates P, if r being in the Gulf Coast implies P(r), then P(r).
$\forall r,P\,[R(r) \wedge (G(r) \rightarrow P(r))] \rightarrow P(r)$
Type Of Belief: definitional
Note: This is a second order axiom. It is used to eliminate the Gulf Coast precondition. This precondition is not carried over into NFR12 because it rarely fails, and when it does fail, the effect is not disastrous. The justification records the way in which it is important (*i.e.*, through the default values for distortion region thickness and fault orientation).

B74: The thickness of the distortion region of a Late Fault Region in the Gulf Coast has relation R_1 to the parameter τ.
$\forall r\,[LFR(r) \wedge G(r) \rightarrow R_1(t(dr(r)), \tau)]$
Type Of Belief: definitional

B75: $\tau = 13$
Type Of Belief: statistical

B78: $R_1 = <$
Type Of Belief: theoretical

B76: The orientation of a Late Fault Region in the Gulf Coast has the relation R_2 to the parameter θ.
$\forall r\,[LFR(r) \wedge G(r) \rightarrow R_2(o(r), \theta)]$
Type Of Belief: definitional

B77: $\theta = 75$
Type Of Belief: statistical
B79: $R_2 = <$
Type Of Belief: theoretical

III. REASONING FROM ERROR TYPES AND JUSTIFICATION STRUCTURES

The Critic begins by considering a specific instance in which the user corrects or augments the Performance Program's conclusions. Given any such failure, the first step is to determine specific types of errors in specific rules, that could have produced this failure. For example, if the user deletes a system-generated conclusion, then the rule that suggested this conclusion may be suspected to be in error[4], with the possible error types OverGeneralLeftHandSide and OverSpecificRightHandSide (either of these error types could have led this rule to suggest the incorrect conclusion). Similarly, if the user adds a conclusion, then those rules whose right-hand side mentions that conclusion but which did not trigger are identified as possible errors of type OverSpecificLeftHandSide, while rules that did apply but did not make the indicated conclusion may be suspected of OverGeneralRightHandSide errors. (Of course, another plausible explanation for a failure of this type is absence of an appropriate rule.)

At this point, given a suspected belief (*e.g.*, a rule), and a list of possible error types for the belief, the Critic examines the justification for that belief in order to generate a list of candidate hypotheses regarding possible causes of the error (*i.e.*, bugs in the supporting beliefs, or approximations in the justification links relating the belief to its supporting beliefs). The method for generating and pruning hypotheses about the cause of the error is summarized below:

```
Explain(possible-errors, belief)
begin
  <for each possible error, enumerate supporting
     beliefs that could have caused the error,
     along with their suspected error type>
  <prune these suspects>
    <attempt to determine the correctness of each
     suspect in the current situation, removing
     those shown to be correct.>
    <rank remaining suspects according to their
     Type Of Belief, and remove suspects whose
     Type Of Belief is "definitional".>
  <recursively Explain each remaining suspect>
end
```

This method for identifying suspect beliefs makes use of several kinds of information recorded in the justification structure for the offending rule, as well as knowledge of the context in which the rule failure occurred. These kinds of information and their use may be summarized as follows:

The logical dependencies of one belief on another, recorded in the justification structure, provide the basis for generating candidate suspects. The justification structure is the basis for a "complete" generation of suspects, in the sense that the only possible errors in the underlying domain knowledge that could produce the detected rule error are those involving beliefs mentioned in the rule's justification structure. All the other sources of knowledge that enter into the process serve to constrain the suspects generated on this basis.

The additional knowledge about error types, together with the Error Propagation knowledge associated with each justification link, allows the Critic to prune the set of suspects. In other words, certain supporting beliefs on which the offending rule is based can be pruned as suspects when it can be shown that no error in those beliefs could produce the detected rule error.

Knowledge of the situation in which the failure occurred may allow the system to separately verify the correctness of suspected supporting beliefs. For example, rule NFR12 depends, among other things, upon the assumption that normal faults have associated distortion regions. If this rule is suspected of a failure, that supporting assumption may become suspect. But if there is strong direct evidence in the current situation that a distortion region is present, then that suspect may be pruned.

Type Of Belief knowledge propagated from supporting beliefs as specified by the associated linkage rule allows further ranking and pruning of suspects..

A. EXAMPLE: USE OF THE NFR12 JUSTIFICATION

As an example, we assume a scenario in which the user has indicated that a particular hypothesized "normal fault" should be specialized to a "late fault." The LAS takes this as an indication that it has committed an error of omission by failing to make this specialization on its own, and invokes the Critic to explain the failure.

The Critic begins by examining the reasoning trace of the Performance Program and determines that rule NFR12 could have drawn the correct conclusion, but failed to match. For simplicity we will assume that NFR12 is the only rule that could have drawn the correct conclusion.[5]. The Critic examines the situation in which the rule was attempted and, through interaction with the user determines that: *(i)* the user agrees that the normal fault conclusion drawn by the Performance Program is correct; and *(ii)* the Performance Program detected a satisfactory red pattern that is above the fault plane pattern of the normal fault, but is longer than 50 ft. In general, the Critic uses a combination of the reasoning trace and the rule justifications to track down the source of the failure. If, for example, the red pattern had been found to be unacceptable to the user, then attention would have been focussed on the detector for that pattern.

As a result of this preliminary analysis, the Critic hypothesizes that rule NFR12 has committed an error of type OverSpecificLeftHandSide, specifically in clause 2. The Critic now attempts to explain this error in terms of the justification for NFR12 (*i.e.*, for **B1** in Figure II-2), in order to determine which types of errors in its supporting beliefs could have caused this error. The following paragraphs summarize the generation and pruning of suspect beliefs generated by the procedure **Explain** described above.

In the first step of this procedure, the justification of **B1** is examined to find that it depends upon **B2**, and that the OverSpecific LeftHandSide error in **B1** can only be explained by an Over-

[4]Of course it is possible that this rule is correct, that it should not have been applied, but that its preconditions were satisfied only because of an error in an earlier rule. Thus, in general there will be several alternative rule suspects. The Critic weighs the alternative suspect rules by examining their justifications, and determining the plausible causes of errors for each.

[5]Of course if other rules are available which could have drawn the correct conclusion, then their justifications must be examined to generate additional hypotheses. While this may add substantially to the number of hypotheses that the Critic must consider, it does not change the reasoning that the Critic goes through in considering each hypothesis.

SpecificrightHandSide error in **B2**, as a result of the Abductive linkage rule.

The Error Propagation information associated with **B1** follows from the linkage rule, and is used to determine the error type for **B2**.

By propagating the error in **B1** in this way, the problem of explaining the OverSpecificLeftHandSide error in **B1** is reduced to the problem of explaining the corresponding OverSpecificRightHand-Side error in **B2**; specifically, in clause 2 of **B2**. Note once more that the exact identification of the the error type for **B2** is determined by a combination of propagating the error type from **B1**, and checking the left-hand side of the implication **B1** against the known facts in the current situation.

The OverSpecificRightHandSide error in **B2** is explained in turn by examining its justification. As shown in the figure, **B2** is derived from **B3** and **B4**. This step in the justification corresponds to a renaming of terms, and the beliefs themselves are not particularly interesting. However, one important point is worth noting: while the error in **B2** can be explained by an OverSpecific-RightHandSide error in either **B3** or **B4**, the first of these suspect beliefs is eliminated because its Type Of Belief is *definitional* – hence above suspicion.

The remaining error hypothesis, that **B4** has committed an Over-SpecificRightHandSide error, is next propagated to its supporting beliefs. Although any of the supporting beliefs could lead to an OverSpecificRightHandSide error in **B4**, **B7** is pruned because its Type Of Belief of **B7** is *definitional*.

The Type Of Belief of **B8** is *theoretical* (*i.e.*, it can only be incorrect if the approximate domain theory is incorrect). Our current Critic ranks suspects that would require a change to the approximate theory lower in likelihood than suspects that would not require such a change. It will only consider making a change to the approximate theory if it cannot explain the failure in any other way. As a result, **B8** is noted for later examination if another explanation cannot be found. As a result of this pruning, the Critic focusses its attention on an OverSpecificRightHandSide error in **B6**.

The further error propagation from **B6** to its supporting beliefs illustrates the importance of distinguishing error types to the task of eliminating suspects. In this case, while there are five beliefs that directly support **B6**, only two of these become suspects for explaining the detected error. This is because the error propagation information associated with **B6** indicates that no kinds of errors in the other three beliefs could result in the present error. Thus, the two candidate suspects at this point are OverSpecific-RightHandSide errors in **B65** and **B72**.

Continuing in this manner, the Critic finally arrives at two plausible explanations for the original rule error, corresponding to nodes **B75** – an underestimate of the distortion region thickness and **B77** – an understimate of the orientation. In order to resolve between these hypotheses, the Critic will attempt to observe or infer the orientation and distortion region thickness from other case-specific data. If this course of action fails, it will rank these suspects hypotheses based on the relative Degree of Belief of the two default value assumptions.

Now that the Critic has isolated the possible explanations of the detected error, a decision must be made regarding whether and how to alter the offending rule. In cases where the explanation indicates that the training instance is a statistical outlier (*i.e.*, that the orientation is several standard deviations above the mean in this case), it may be best to leave the rule as is and simply update the statistics supporting the default value of the orientation. If the current case suggests instead that the default value is often incorrect (*e.g.*, that in fact in most cases the assumption is violated), then the default may be changed and the rule revised accordingly (*e.g.*, change the maximum length of the red pattern in this rule to a larger number). In general, the decision of whether and how to change the rule must depend on a cost benefit analysis which takes into account benefits of changing the rule (*e.g.*, increased accuracy) as well as the costs (*e.g.*, referring to parameters such as orientation which may not often be known, and which therefore make the rule unusable in many cases). At present we plan for an interactive rule editor that will present the results of the Critic's analysis to the user to consider alternative rule repairs.

IV. CONCLUSIONS AND DIRECTIONS

We have shown the utility of a particular linkage of shallow rules to underlying domain knowledge as an aid to knowledge base refinement in the context of a Learning Apprentice System. The linkage called a justification structure, explicitly records the assumptions and approximations involved in the derivation of a shallow rule. Furthermore, we have demonstrated the utility of maintaining a taxonomy of error types. Information about the type of error that has been observed, together with the justification structure, can help focus the Critic's search for plausible suspect beliefs.

The work reported here represents the initial results of our efforts to explore the representation and use of justifications to support automatic refinement of knowledge bases. While several recent research efforts have focused on constructing explanations, or justifications, to guide generalization learning [4, 6, 12], this work complements those efforts by considering complex explanations involving default assumptions, and non-truth preserving justification rules.

While the importance of constructing and using justifications in learning is clear, there are several serious issues that remain to be understood. These include:

It is still not clear precisely what statement about a rule should be justified. In this paper the focus is on justifying the *logical correctness* of the rule. However, in many cases the *Dipmeter Advisor* system rules are not, strictly speaking, correct. Instead, they are convenient approximations to the truth. For example, the rule NFR12 is not correct since its preconditions are not logically sufficient conditions for assuring the existence of a late fault. To be correct it would require many more preconditions, including the condition that the borehole is not washed out, and that the fault angle is less than 75°. Unfortunately, such a logically correct version of the rule would not be very useful because one rarely knows enough about a given situation to assure that all the necessary preconditions actually are satisfied (*e.g.*, one typically does not know the angle of the fault). Therefore, we create and use rules that are convenient falsehoods – simple enough to be useful in a broad range of

situations, and close enough to the truth to keep from creating disaster when they are incorrect. An appropriate justification for rules should therefore take into account tradeoffs between correctness, applicability, cost of application, cost of different types of failure, and utility of a rule. We intend to explore various criteria for characterizing the appropriateness of rules, and to extend our formalism to accommodate these criteria.

We have only developed weak methods for dealing with the expected interaction of rule refinement and extension of the underlying domain theory. We have assumed, for the most part, that the underlying theory is correct and that failures are due to violated assumptions and approximations used in deriving the rules. When this is not true, our only recourse is to engage in a dialogue with the specialist.

Another fundamental issue lies in the difficulty of acquiring rule justifications. The justification of NFR12 involves some 65 supporting beliefs. Acquisition of large numbers of justifications with this complexity, along with acquisition of the rules themselves, could place a significant additional burden on our domain specialists. Such a burden could have serious negative implications for practical use of justifications in knowledge base refinement. We have, however, succeeded in constructing a prototype system for generating interpretation rules from an approximate domain theory, and associated procedures for constructing the rule justifications based on the trace of the generation procedure. Although this is provocative and encouraging, we do not yet understand the complexity of justifications, and we have much to learn about the difficulties of constructing complex justifications in a domains with imperfect theoretical underpinnings (such as geology).

ACKNOWLEDGEMENTS: John McDermott, Smedar Kedar-Cabelli, and Rich Keller made a number of insightful suggestions and comments on an early draft of this paper. Mike Barley also tested some of the ideas. The reviewer offered a number of extremely helpful criticisms and suggestions.

REFERENCES

1. Buchanan, B.G., and E.H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1984.

2. Buchanan, B. G., Mitchell, T. M., Smith, R. G. and Johnson, C. R. Jr. "Models of Learning Systems." *Encyclopedia of Computer Science and Technology 11* (1978), 24-51. also Stanford report STAN-CS-79-692

3. R. Davis. *Applications Of Meta-Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge Bases.* Ph.D. Th., Stanford University, July 1976. STAN-CS-76-552

4. G. DeJong. Acquiring Schemata Through Understanding And Generalizing Plans. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, August, 1983, pp. 462-464.

5. J. Doyle. "A Truth Maintenance System." *Artificial Intelligence 12*, 3 (1979), 231-272.

6. T. M. Mitchell. Learning And Problem Solving. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, August, 1983, pp. 1139-1151.

7. H. E. Pople, Jr. On The Mechanization Of Abductive Logic. *Proceedings of the Third International Joint Conference on Artificial Intelligence*, August, 1973, pp. 147-152.

8. H. E. Shrobe. *Dependency Directed Reasoning for Complex Program Understanding.* Ph.D. Th., MIT, April 1979. (AI-TR-503)

9. R. G. Smith and R. L. Young. The Design Of The Dipmeter Advisor System. *Proceedings of the ACM Annual Conference*, ACM, New York, October, 1984, pp. 15-23.

10. W. R. Swartout. "XPLAIN: a System for Creating and Explaining Expert Consulting Programs." *Artificial Intelligence 21* (1983), 285-325.

11. H. A. Winston, R. G. Smith, T. M. Mitchell, and B. G. Buchanan. *A Complete Rule Justification Example.* Research Note , Schlumberger-Doll Research, August, 1985.

12. P. H. Winston, T. O. Binford, B. Katz, and M. Lowry. Learning Physical Descriptions From Functional Definitions, Examples, And Precedents. *Proceedings of the National Conference on Artificial Intelligence*, August, 1983, pp. 433-439.