Context Refinement – Investigating the Rule Refinement Completeness of SEEK/SEEK2

Hans-Werner Kelbassa [1]

Abstract. The pioneer systems for rule refinement are SEEK and SEEK2. Unlike TEIRESIAS, which also has been designed for the acquisition of new inference rules, the systems SEEK and SEEK2 are devoted to the refinement of rules for a rheumatology rule-base (a medical diagnosis application). This article investigates the general refinement completeness of SEEK/SEEK2. A rule refinement system is complete if it solves every possible refinement problem. SEEK2 has refinement heuristics for coping with generalization and specialization problems. Complete rule refinement systems should also have refinement capabilities for tackling a third refinement class to be called context refinement. On the syntactic level, the rheumatology rules which are the subject of the SEEK/SEEK2 refinements have no logical negation in their if-parts. On the semantic, findings which are to be interpreted by the rheumatology rule-base are represented in positive form only. This seems to be the reason for the incompleteness of SEEK2 with regard to context problems, i.e., there was no need for context refinement heuristics. A complete rule refinement system must employ methods for contextualization, as well as generalization and specialization.

1 INTRODUCTION

Today there is no complete rule refinement system which can execute all possible rule refinements in an optimal manner [2]. The refinement systems SEEK and SEEK2 are the pioneer systems for rule refinement. Both systems were developed at Rutgers University (New Jersey). The system SEEK was completed in 1982 by P. Politakis [12]; the successor system SEEK2 was finished in 1988 by A. Ginsberg [6]. The application of SEEK/SEEK2 was the refinement of rules for the diagnosis of rheumatological diseases, which were written using the system EXPERT. The key strategy for the development of this rule-based expert system for rheumatology was the testing of the expert system conclusions against an available data base of clinical cases with known diagnoses.

However, due to the applicability of SEEK the expert system for rheumatology, within relatively few years, got a large rule-base: "The model has been critiqued by an external panel of expert rheumatologists, and a review of performance has shown the diagnostic accuracy in 95% of 145 clinical cases ...... At this time, the dimensions of the model include 30 final diagnostic conclusions, 600 intermediate conclusions, 900 observations, and over 1000 production rules." [2]

The SEEK system succeeded by performing two kinds of heuristic refinement: rule generalizations and rule specializations. Generalizations are refinements that weaken a rule, resulting in a new rule which logically includes the previous one. Specializations are rule refinements which, strengthening a rule, result in a new rule which logically includes the previous one. A generalized rule fires more often than the previous one did. A specialized rule fires less often than the refinement candidate did. An example for a rule generalization is the removal of a condition from the conjunctive if-part of this rule. A rule specialization is the inverse refinement operation, for example, the insertion of an additional condition in the conjunctive if-part of any rule.

The generalization or specialization of a rule can be visualized by an inference table, showing the different microstates in which the rule does or does not fire.

Table 1. Inference table for the production rule $(A \wedge B) \rightarrow I$ with validation states demanding a rule generalization.

Micro State	A	B	$A \wedge B$	Inference XPS	Inference EXP	Validation State
1	0	0	0	0	0	valid
2	1	0	0	0	0	valid
3	0	1	0	0	1	falsifi ed
4	1	1	1	1	1	valid

Table 1 presents the microstates for the production rule $R := (A \wedge B) \rightarrow I$. The comparison of the inferences from the expert system (XPS) and the domain expert (EXP) shows that microstate 3 was falsified, because the rule did not fire. In this example the domain expert is demanding a rule generalization, i.e. the rule should fire in addition in microstate 3, too. The target rule for this refinement is $R^* = (A \wedge B) \vee (\neg A \wedge B) \rightarrow I$; this rule is equivalent to $R^* = B \rightarrow I$.

In order to derive the right refinement for a certain rule, SEEK gathered rule performance statistics to be considered as meta knowledge. This performance information reveals how often the considered rule has been fired in the right way and how often it was a wrong or missing element in the rule trace for several cases. Based on this meta knowledge SEEK could generate suggestions for the correction of misdiagnosed cases, i.e., SEEK gave interactive advice about rule refinement during the design of the rule-based consulting system for rheumatology.

The technical term *rule refinement* distinguishes the initial rule-base construction phase from the rule revision phase to be performed later on, after the expert system has reached a considerable competence, but is not always able to find the valid expert reasoning paths. There are failures which should not lead to the removal, but to the correction of faulty rules. The central idea of refinement systems is to find a *minimal* revision, so that the falsified cases get valid reasoning paths without any side effects. It is assumed that the number of

[1] Email: kelbassa@uni-paderborn.de
[2] Politakis [12], p. 8

rules does not change during the refinement phase.

The difference between SEEK and SEEK2 is the degree to which the user interacts with the refinement system. SEEK2 has an automatic refinement capability, therefore it can perform basic tasks without expert interaction: "The output of SEEK2 running in automatic mode is not a list of suggested rule refinements ....., rather it is a refined version of the entire knowledge base, i.e., a set of rule refinements to the initial knowledge base which yield an improvement in overall performance." [3]

In order to examine the general refinement power of SEEK/SEEK2, it is to be investigated which kind of rules have been refined by these systems [7]. The main focus of this article is to find out whether the refinement capability of SEEK/SEEK2 is complete, or whether due to the special rule representation there is any 'refinement gap' to be coped with by additional research.

The next section characterizes the EXPERT rule representations SEEK/SEEK2 dealt with [13]. Then the rheumatology rules are described and analyzed. We show that SEEK/SEEK2 rules are not representing logical negations by symbols as 'NOT', but by negative confidence values. Moreover, we show that the SEEK/SEEK2 rules have a representation gap with regard to the logical negations of conditions in the if-parts. So it is no wonder that SEEK/SEEK2 cannot cope with a third rule refinement category to be called contextualization. This kind of rule refinement is defined and described in section 4. Then the validation measures of SEEK2 are examined.

2 TABULAR FORMAT FOR SEEK's RULES

The rules being refined by SEEK/SEEK2 have been written using the EXPERT rule representation language. One particularity of EXPERT is the n-of-m rule representation, i.e., the rule interpreter fires this kind of *'choice rule'* if $n \in I\!N$ or more of its $m \in I\!N$ conditions are satisfied ($n \leq m$). This choice rule led the developer of the rheumatology rule-base to the processing of criteria tables which list the possible major and minor findings (observations, user inputs) concerning a certain disease.

Table 2 presents the major and minor criteria for the diagnosis of mixed connective tissue disease (MCTD).

Table 2. Major and minor findings for mixed connective tissue disease (MCTD) diagnoses (Politakis 1985).

Major Criteria	Minor Criteria
1. Swollen hands	1. Myositis, mild
2. Sclerodatyly	2. Anemia
3. Raynaud's phenomenon or esophageal hypomotility	3. Pericarditis
4. Myositis, severe	4. Arthritis $\leq$ 6 wks
5. codiff capacity, nl :$<$ 70	5. Pleuritis
	6. Alopecia

The elements of the criteria table may be intermediate results obtained by reasoning rules expressed in other tables. For example, the minor *Pleuritis* in the criteria table above may be derived by tabular model rules for reaching this conclusion. It is also possible that major criteria of the criteria table may be derived by inference rules. The tabular model is a special kind of rule representation supported by the EXPERT rule representation language. Table 2 shows the tabular format of the rules for diagnosing mixed connective tissue disease (MCTD) representing three confidence levels: definite, probable, and possible.

[3] Ginsberg et al. [8], p. 207

Table 3. Tabular format of SEEK/SEEK2 rules for mixed connective tissue disease (MCTD) diagnoses (Politakis 1985).

Confidence Level	Definite	Probable	Possible
Major Criteria Minor Criteria	4 Majors	2 Majors 2 Minors	3 Majors
Requirements	Positive RNP Antibody	Positive RNP Antibody	No requirement
Exclusions	Positive SM Antibody	No exclusion	No exclusion

The terms 'Majors' and 'Minors' in the above tabular format for MCTD refer to the criteria table as shown above. In this tabular format for MCTD each column represents a rule for a certain confidence level. The row for requirements lists important basic requirements to be satisfied for a special diagnosis. The row for exclusions in this tabular format lists those observations which rule out the diagnosis (here MCTD) at the indicated confidence level. Interpreting the tabular format for mixed connective tissue disease (MCTD) on the definite level leads to the following MCTD rule:

IF the patient has 4 or more major findings for MCTD,
AND RNP antibody is positive,
AND SM antibody is not positive,
THEN conclude definitive MCTD.

This MCTD rule is a typical SEEK rule. The rules to be refined by SEEK/SEEK2 obtain their logical negations from the row for exclusions. The major and minor criteria are to be considered as 'positive' conditions for a certain disease, but if the exclusion condition appears, the expert system is unable to conclude this disease for the given case, i.e. the exclusion is a 'negative' condition for a special disease which rules out the rule conclusion. As 'Positive SM Antibody' is an exclusion for MCTD, it appears in its negative form in the above sample rule.

3 RULE REPRESENTATION ANALYSIS

The focus of the representation analysis to be done here is the evaluation of the logical power of the rules which are subject of the SEEK/SEEK2 refinements. If there is any *'representation gap'* with regard to the power of first-order logic rule representation languages, the refinement efforts of SEEK2 cannot be subject of the missing rule representation form. In this situation SEEK/SEEK2 are incomplete concerning the performed rule refinement spectrum.

The representation problem can be analyzed by using the following SEEK2 rule [8]:

IF 2 or more of the following conditions are satisfied:
Hypothesis NEPH has confidence between 0.9 and 1
Finding MALAR is true
Hypothesis SEROS has confidence between 0.9 and 1
Hypothesis CNS has confidence between 0.9 and 1
Finding HEMAN is true
AND 2 or more of the following conditions are satisfied:
Finding FEV is true
Finding ARTH is true
Finding GGLOB is greater than 1.8
Hypothesis HCMP has confidence between 0.9 and 1
Finding PLAT is less than 100
AND Hypothesis RD203 has confidence between 0.9 and 1
AND Hypothesis EX1SL has confidence between -1 and 0.05
THEN Conclude Hypothesis SLE with confidence 0.9

The first two conditions are n-of-m conditions as mentioned above. The third condition is a normal condition with positive confidence values. But the fourth condition, referring to the exclusion for the disease, has an exception: it shows that the exclusion is not processed on the logical level, rather on the confidence value level. If the exclusion EX1SL had been processed on the logical level, the condition would have been represented as '(NOT (Hypothesis EX1SL))' - or using any other symbol for '$\neg$'. But the SEEK2 rule has characterized the exclusion by negative confidence. SEEK/SEEK2 confidence values are assigned to conclusions with the numeric range of -1 and +1, whereby the value -1 indicates complete denial, and the value +1 complete confirmation.

So the representation analysis of SEEK2 rules reveals on the syntactic level that these rules have no logical negation, i.e. the symbol 'NOT' is not present in any exclusion condition. On the semantic level it is to be ascertained that the exclusions are not referring to the findings. This means that the elements of the criteria tables and the exclusions are *disjoint* sets:

$$(criteria\ table\ sets) \cap (exclusion\ sets) = \emptyset.$$

Moreover, there is no finding F that appears as (NOT F) in the if-part of any rule. As the findings are not elements of the exclusion sets, they are not appearing in negative conditions on the logical level. There seems to be no need for SEEK2 to cope with this kind of refinement problem. The tables 2 and 3 above show the criteria table and the tabular format for MCTD. Comparing the major and minor criteria in table 2 with the exclusion set for MCTD in table 3, we see that the exclusion 'Positive SM Antibody' is not an element of the major/minor criteria table for MCTD. Therefore, this comparison confirms the result that the exclusion sets and the criteria table sets are disjunct ones. This outcome means that there is a representation gap regarding the *explicit* negation of major/minor observations; the SEEK2 rules which are subject to refinement do not explicitly represent any negated major or minor observations.

In OPS-5 there is an action REMOVE which deletes elements from the working memory [4]. REMOVE-related rule refinements are not carried out by SEEK2.

4 CONTEXT REFINEMENT

The rule representation problem of SEEK2 can be explained by using the following inference table for the simple production rule $R := (A \wedge B) \rightarrow I$ showing two falsified microstates.

Table 4. Inference table for the rule $(A \wedge B) \rightarrow I$ with two falsified validation states characterizing a context problem.

Micro State	A	B	$A \wedge B$	Inference XPS	Inference EXP	Validation State
1	0	0	0	0	0	valid
2	1	0	0	0	0	valid
3	0	1	0	0	1	falsified
4	1	1	1	1	0	falsified

The inference table shows the 4 microstates for the rule R, and the inferences of the expert system (XPS) and the expert (EXP). The difference between the inferences obtained from the expert and the expert system leads to the falsified microstates 3 and 4. In this example the microstate 4 is falsified because the domain expert does not accept the firing of rule R. Instead of this the domain expert is demanding a rule which fires in microstate 3. The refined rule for this example is $R^* = (\neg A \wedge B) \rightarrow I$. Table 4 shows that this refinement is a 'context switching': the refined rule will no longer fire in the old context of microstate 4, rather this rule will fire in the context of microstate 3, i.e. the old rule firing context will disappear and the new one will appear. This rule refinement is *neither* a generalization *nor* a specialization [6]. It belongs to a *third* refinement class called *context refinement* [9] because it will happen that the refined rule will fire in *another context*. The same is true if the rule $R2 := (A \vee B) \rightarrow I$ is refined by inserting a negation. If, for example, the revised rule is $R2^* = (\neg A \vee B) \rightarrow I$, then this rule will not fire more or less than R2 did.

The problem with SEEK2 is that the refinement dichotomy – generalization and specialization – is not complete because there is a third refinement class: context refinement (contextualization) which is defined in the following way:

> *Context refinement (contextualization) is a rule refinement class which is neither a generalization nor a specialization. The application of a context refinement to rule R always results in a revised rule R^*, so that neither R^* logically includes R nor R logically includes R^* : $R \cap R^* = \emptyset$.*

Whether a target refinement belongs to the refinement class generalization or specialization can be determined by the interpretation of the rule's validated microstates presented in an inference table. Table 4 shows that for context refinement there is no logical intersection of the rules R and R^*, because $R \cap R^* = \emptyset$ holds. In table 4 the microstate 4 fires, in the revised rule R^* the microstate 3 fires. So there is no logical intersection: $R \cap R^* = \{4\} \cap \{3\} = \emptyset$.

Context refinement is important because many tools (e.g. KEE) represent logical negations by 'NOT', so that the rule base to be refined has rules of the form IF (A OR (NOT B)) THEN I.

For generalization the following definition holds:

> *Generalization refinement (generalization) is a rule refinement class that consists of refinements whose application to rule R always results in a revised rule R^* which logically includes R, so that it is logically impossible for the premises of rule R to be satisfied without the premises of rule R^* being satisfied: $R \subset R^*$.*

For premise refinement let P be the premise(s) of rule R and let P^* be the premises of the revised rule R^*. If the generalization of R was performed by changing the premises only, $P \subset P^*$ is present whereby $P \subset P^*$ denotes the *logical inclusion*, i.e. the set of microstates fulfilling P^* is a superset of the set of microstates fulfilling P – and therefore $R \subset R^*$ holds, too. [4] This definition by Ginsberg also regards a refinement that keeps the rule's premises constant but allows it to reach a stronger conclusion in the logical sense [6]. If the generalization of rule R was reached by changing the conclusion(s) C into C^* only, the relation $C \subset C^*$ holds. This means with regard to the confidence level of SEEK2 that a generalization effects a higher confidence level.

For specialization the following definition [6] holds:

> *Specialization refinement (specialization) is a rule refinement class that consists of refinements whose application to rule R always results in a revised rule R^* which logically includes R, so that it is logically impossible for the premises of rule R^* to be satisfied without rule R being satisfied: $R^* \subset R$.*

[4] See the generalization example in table 1: $R \subset R^*$ holds, because of R the microstate 4 fires and of the refined rule R^* the microstates 3 and 4 fire. So we get $R \subset R^* = \{3\} \subset \{3, 4\}$

If the specialization of rule R was performed by changing the premises only, then $P^* \subset P$ holds whereby $P^* \subset P$ denotes the *logical inclusion*, i.e. the set of microstates fulfilling P^* is a subset of the set of microstates fulfilling P – and therefore $R^* \subset R$ is valid. But if the rule was specialized by changing the conclusion(s) only, then $C^* \subset C$ holds, i.e. the specialized rule R^* has a weaker conclusion.

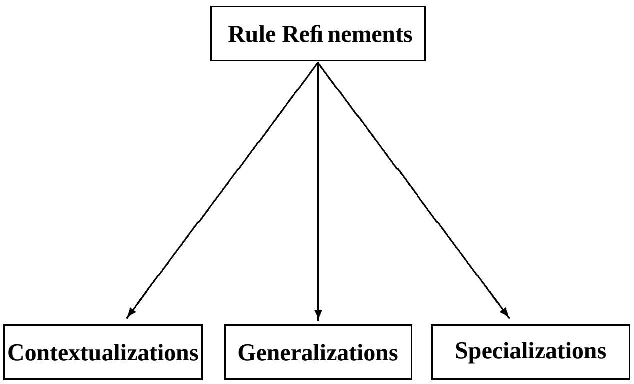

Figure 1. Rule Refinement Trichotomy.

An important part of rule representation is the attribute-value condition in the if-part of rules [1]; for example

IF Attribute B has value x, THEN Conclude hypothesis Z.

This kind of rule can be refined by changing the symbolic value, for example, by the substitution of value x for value y:

IF Attribute B has value y, THEN Conclude hypothesis Z.

This rule refinement belongs to the class context refinement because it is not a generalization and also not a specialization [6].

The systems SEEK/SEEK2 are working with known input – output relations, i.e. the case base represents for every case the patient findings (observations) to be considered as user input and the known final clinical (rheumatology) diagnosis. SEEK2 was not designed to cope with intermediate rule problems: "What matters is the knowledge base's answer, not how it reached it." [5] So, for example, in order to specialize a faulty rule SEEK2 cannot add a component to this rule [8]. However, the reason for these shortcomings is that SEEK2 has no acquisition facility and no *validation interface* [9]. Therefore SEEK2 is unable to identify context problems regarding intermediate rules as defined above because it cannot acquire the refinement target information or the *failure critique* by the domain expert.

If a SEEK2-like system should add a component to the conjunctive if-part of a rule (specialization), the system known conditions should be shown to the validating domain expert so that he or she can add the right condition to the rule by mouse-click-techniques. This kind of validation interface has been described in [9]. The central idea of TEIRESIAS was *single case analysis* by rule tracing [3]. The central idea of SEEK2 is *multiple case analysis* and *automatic* refinement. So the domain expert should be able to enter recognized reasoning failures or missing and wrong outputs by a validation interface. If this is done in a TEIRESIAS-like manner case by case, the rule validation module can gather rule *performance statistics* in a SEEK2-like manner. The idea pursued here is to acquire as much as possible *validation expertise* from the domain expert during the evaluation of the expert systems reasoning path and to look for excellent *validation measures* that characterize the need for a special refinement with high certainty. The rule refinement proposed by the validation system need not be accepted by the domain expert; he can reject (empirical evaluation). The domain expert is the final judge.

[5] Ginsberg [6], p. 8

5 VALIDATION MEASURES

For the performance of rule refinement heuristics it is crucial to find well-defined measures for the different validation problems. Some problems concerning rule validation measures will be described now by using simple examples.

If the rule $(A \wedge B) \rightarrow I$ is fired and the validating expert rejects this system result, because the correct rule should be $(A \wedge \neg B) \rightarrow I$, SEEK2 will register a need for specialization, although the target refinement is context refinement. The statistical measures, which SEEK2 invokes if the fired rule is *not* the right one, are the measures $SpecA(R_x)$ and $SpecB(R_x)$ with R_x meaning any rule of the given rule base RB ($R_x \in RB$).

The SEEK2 validation measure $SpecA(R_x)$ is defined as follows:

$SpecA(R_x)$ is the number of cases in which
(a) this rule's conclusion should not have been reached but was, and
(b) if this rule had failed to fire, the correct conclusion would have been reached.

If there is more than one fired rule that concludes the incorrect result, none of these rules will have its SpecA measure incremented. *Instead of this* SEEK2 *has an additional concept to cover this situation called $SpecB(R_x)$: each of these faulty rules get its SpecB measure incremented.*

However, the SpecA and the SpecB measures register context problems inadequately. A context problem is *neither* a generalization *nor* a specialization. SEEK2 has no validation measure which is regarding context refinement [6]. Therefore, there is a need for a new measure. This becomes obvious when the $Gen(R_x)$ measure is examined now.

The above example with the SpecA measure interpreted an explicit inference failure. In addition, there are implicit inference failures, too. If the expert system did *not* fire the rule $(A \wedge B) \rightarrow I$ because the if-part of this rule is not satisfied, the target refinement can, for example, get the rule $(A \wedge \neg B) \rightarrow I$. This means that a context refinement is required in order to cope with this implicit inference failure. In this situation SEEK2 invokes the validation measure $Gen(R_x)$ for the missing rule trace element $R_x \in RB$.

The validation measure $Gen(R_x)$ is defined in the following way:

$Gen(R_x)$ is the number of cases in which
(a) this rule's conclusion should have been reached but was not,
(b) had this rule been satisfied, the right conclusion would have been reached, and
(c) of all the rules for which the preceding clauses hold in this case, this one is the closest to being satisfied.

The SEEK2 system registers the implicit inference failure as generalization problem, although there is a context refinement on target. So the $Gen(R_x)$ measure for rule validation is inadequate, too.

A basic heuristic of SEEK2 is the generalization heuristic [8]:

$$Gen(R_x) \ > \ Spec(R_x).$$

If the value of $Gen(R_x)$ is larger than the value of $Spec(R_x) \ = \ SpecA(R_x) \ + \ SpecB(R_x)$, rule R_x will become a generalization candidate. The idea of this heuristic is excellent, but this heuristic is unable to cope with context problems. The measures for generalization and specialization do not have to cover context problems, i.e. the measure for context refinements must be separated from the registration of the *genuine* generalization and specialization problems. This is already possible during the evaluation session after the validation interface has acquired the *target refinement*. However, this topic is interesting for future rule validation research. It will be described now in which manner a validation measure for context problems can be applied.

Let $Con(R_x)$ be the number of cases, in which context refinement is required so that the falsified cases get valid reasoning paths. This validation measure must enable the validation module to decide which kind of refinement is to be performed for rule R_x. Therefore, the target refinement class for any rule R_x must be selected on the basis of the maximum of the functions $Con(R_x),\ Gen(R_x),\ and\ Spec(R_x)$. *Let $Tar(R_x)$ be the refinement class which is to be determined as the target one for rule R_x.*

Then $Tar(R_x)$ is defined by the following expression:

$$Tar(R_x) \ := \ max\{Con(R_x),\ Gen(R_x),\ Spec(R_x)\};$$
$$Con(R_x) \in I\!N,\ Gen(R_x) \in I\!N,\ Spec(R_x) \in I\!N.$$

This new measure will enable the validation module to process context refinements in a SEEK2-like manner. In order to constitute a numeric example, let the value of $Con(R_{64})$ be 110, let the value of $Gen(R_{64})$ be 28, and let the value of $Spec(R_{64})$ be 70; then the following result is obtained: $Tar(R_{64}) = max\{Con(R_{64}), Gen(R_{64}), Spec(R_{64})\} \ = \ Con(R_{64})$. This means that rule R_{64} must be improved by context refinement and the expected validation gain is 110 cases.

6 CONCLUSION

This article is a contribution to the development of complete rule base refinement systems which acquire validation expertise and use it to suggest rule modifications [9]. For the time being there is no testing of context refinement heuristics because this generic refinement class has not been defined elsewhere [8]. It is not yet clear whether the three rule refinement categories contextualization, generalization, and specialization are containing all possible rule refinements. So this trichotomy is to be considered as a minimal rule refinement classification.

SEEK2 can perform rule generalizations and rule specializations only. The rules refined by SEEK2 have been written using the tool EXPERT. Unlike rules processed by other tools, e.g. KEE, an EXPERT rule base does not include rules of the form IF (A OR (NOT B)) THEN X. If this sample rule is to be refined into IF (A OR B) THEN X then SEEK2 is unable to execute this refinement because there are no measures and heuristics for context refinement. However, there are many real world rule bases demanding that a refinement system must cope with this *'normal'* kind of rule refinement.

The SEEK2 idea of gathering rule meta knowledge and using refinement heuristics is a generic approach to designing rule refinement systems [5]. Unfortunately, the SEEK2 measures for generalization and specialization cannot be the basis for a complete rule refinement system because these measures fail if context problems appear as defined above. This refinement category contains an important kind of symbolic attribute-value refinement which for practical reasons cannot be ignored. One characteristic of context refinements is that the present firing microstate of the rule disappears and that a new one will appear.

The reason for the shortcomings of SEEK2 is that SEEK2 has no acquisition facility and no *validation interface* (section 4). Therefore SEEK2 is unable to identify context problems for intermediate rules because there is no possibility of acquiring the refinement *target* information or the *failure critique* by the domain expert. So the need for a rule validation interface has been established. Such kind of validation interface will enable a merger of the design ideas of TEIRESIAS and SEEK2.

In order to design a general rule refinement system there must be a more powerful case-based approach for the refinement heuristics, too. In [10] it is ascertained that SEEK2-like first order refinement heuristics are suboptimal for cases with multiple refinement problems and that there is a need for higher order refinement heuristics for coping with this problem. It is to be emphasized that it is possible to perform a *mathematical optimization* for the final rule refinement selection stage [11]. Moreover, as uniform case weights are not sufficient, a generic rule refinement system should employ a concept of case weighting so that heterogeneous cases get different case weights.

REFERENCES

[1] R. Boswell and S. Craw, 'Organizing knowledge refinement operators', in *Validation and Verification of Knowledge Based Systems - Theory, Tools and Practice.*, eds., A. Vermesan and F. Coenen, Proceedings 5th European Symposium on Verification and Validation of Knowledge Based Systems and Components (EUROVAV '99), pp. 149–161, London, (1999). Kluwer Academic Publishers.

[2] R.A. Boswell, 'Knowledge refinement for a formulation system', Technical report, Ph.D. Dissertation. The Robert Gordon University,School of Computer and Mathematical Sciences, Aberdeen, U.K., (1999).

[3] R. Davis, 'Interactive transfer of expertise: Acquisition of new inference rules', *Artificial Intelligence*, **12**, 121–157, (1979).

[4] C.L. Forgy, 'Rete: A fast algorithm for the many pattern / many object pattern problem', *Artificial Intelligence*, **19**, 17–37, (1982).

[5] A. Ginsberg, 'A metalinguistic approach to the construction of knowledge base refinement systems', in *Proc. 5th National Conference on Artificial Intelligence 1986 (AAAI '86)*, pp. 436–441, (1986).

[6] A. Ginsberg, *Automatic Refinement of Expert System Knowledge Bases*, Pitman Publishing, London, 1988.

[7] A. Ginsberg, S. Weiss, and P. Politakis, 'Seek2: A generalized approach to automatic knowledge base refinement', in *Proc. 9th International Joint Conference on Artificial Intelligence 1985 (IJCAI '85)*, pp. 367–374, (1985).

[8] A. Ginsberg, S.M. Weiss, and P. Politakis, 'Automatic knowledge base refinement for classification systems', *Artificial Intelligence*, **35**, 197–226, (1988).

[9] H.-W. Kelbassa, *Fallbezogene Revision und Validierung von regelbasiertem Expertenwissen fuer die Altlastenbeurteilung*, 276–285, Informatik-Fachberichte 256, Springer-Verlag, Berlin, 1990.

[10] H.-W. Kelbassa, 'Higher order rule refinement heuristics for rule validation', in *Proceedings 15th Florida Artificial Intelligence Research Society Conference 2002 (FLAIRS-2002). Pensacola, Florida: May 14-16*, (2002).

[11] H.-W. Kelbassa, *Selection of Optimal Rule Refinements*, Forthcoming, Paderborn, Germany, 2002.

[12] P.G. Politakis, *Empirical Analysis for Expert Systems*, Pitman Publishing Inc., Boston, 1985.

[13] S. Weiss and C. Kulikowski, 'Expert: A system for developing consultation models', in *Proceedings 6th International Joint Conference on Artificial Intelligence 1979 (IJCAI '79)*, pp. 942–947, (1979).