# The Rule Retranslation Problem and the Validation Interface

**Hans–Werner Kelbassa**
Schledebrückstraße 112
D – 33332 Gütersloh
*Phone:* +49-5241-961770
*kelbassa@upb.de*
Germany

**Rainer Knauf**
Faculty of Computer Science and Automation
Technical University of Ilmenau
PO Box 100565, D – 98684 Ilmenau
*rainer.knauf@tu-ilmenau.de*
Germany

### Abstract

Current rule base maintenance is wasting refinement and inference performance. There are only few maintenance concepts, which enjoy both (1) formal rule refinement and (2) utilizing topical knowledge provided by experts within the refinement process. The current state of the art in rule base validation and refinement reveals that there is no generic validation interface and no optimal rule trace refinement. This paper characterizes two different retranslation approaches for reduced rule bases and proposes a two–step validation process, which combines a case–based approach with a rule trace validation approach.

## Introduction

In (Collet et al. 1990) the authors express a lack of maintenance concepts of the considered expert system developments for environmental protection. A study of the current state of the art of rule validation and refinement reveals that there is no generic validation interface and no optimal rule trace refinement (Zlatareva and Preece 1994).

The validity statements that are the result of the validation of rule–based systems are useful in refining these systems. Thus, refinement technologies close the *validation loop* by generating a modified rule base that obtains a better degree of validity (Knauf 2000; Knauf et al. 2002b). Here, the refinement technology of Knauf is considered in the context of former reduction approaches. The different and competing requirements to the refinement stage reveal the need of a validation interface. As a result of this insight, the authors propose a two–stage validation process that combines both (1) the validation technique based on the evaluated validators' competences as introduced by Knauf (Knauf et al. 2002b) and (2) a trace validation approach that enables the optimization of alternative and competing rule refinements as introduced in (Kelbassa 2003).

To provide an overview on rule base maintenance, this section sketches some historical milestones. The first rule refinement/revision systems were *single case analysis systems*, which fix a rule base with respect to a concrete case. The drawback of this approach is that the impact of rule refinements on other cases is not evaluated. This insight led to the development of multiple case analysis systems.



## Single Case Analysis Systems

The first rule refinement system is TEIRESIAS (Davis and Lenat 1982). In order to fix recognized reasoning failures, it guides the domain expert through the rule trace interactively and in a case–based manner. The domain expert has the opportunity to modify rules that appeared to be *guilty* in the incorrect processing of the considered case. The problem is that this debugging system is not able to determine the impact of changes within the rule base on other cases. Thus, it might happen that a case, which has been processed correctly before such a modification, is mapped to a wrong solution afterwards. TEIRESIAS does not check for undesired side effects of rule changes and it does not use any rule performance statistics. In particular, the expert is not informed, whether he/she is actually modifying a rule that never failed in the past or a bad one that failed frequently.

In the 1980's, the knowledge acquisition research focused also the development of *intelligent rule editors* like MORE, MOLE SALT, and KNACK (Marcus 1988). MORE, for instance, inspects its current domain model for pairs of hypotheses that do not have any differentiating symptom, i.e. different outputs that are mapped from the same input. If it finds such a pair, the domain expert is asked for a symptom that is able to distinguish both hypotheses (Kahn et al. 1985). There is no proof that these intelligent rule editors control undesired side effects sufficiently.

## Multiple Case Analysis Systems

A pioneer system for this class is the rule refinement system SEEK2 (Ginsberg 1988c), which gathers statistical information on all rules of the validated rule base in order to suggest special refinements to the user. This meta knowledge is processed by heuristics, which enable it to determine whether a rule should be generalized or specialized. SEEK2 can run in either automatic or interactive mode: In the automatic mode, it produces a refined rule base with a better expert system performance. In the interactive mode, SEEK2 gives rule refinement advise to be accepted or rejected by the validator. However, it has been shown, that the refinement dichotomy is incomplete. Besides generalization and specialization there is a third refinement class called *context refinement* (Kelbassa 2002). SEEK2 does not validate the correctness of refined intermediate reasoning chains, but just the right input – output behavior of the system. Thus, the intermediate conclusions might be incorrect. Therefore, this pure single rule validation might not ensure that all refined intermediate rule traces are valid. Since SEEK2 is not a rule trace validation system like TEIRESIAS, it has been proposed to develop a generic validation interface, which can acquire validation information from the evaluating expert(s), for example, knowledge about missing conditions or wrong conclusions in a present rule trace (Kelbassa 1990)[1].

A well-known milestone is the *Reduced Theory Learning System* (RTLS) by GINSBERG (Ginsberg 1988b; Ginsberg 1990). Here, *reduced* means that the RTLS does not refine original rules that also generate intermediate conclusions. Instead, it refines a converted version, which is obtained by a knowledge–base reduction step (Ginsberg 1988a). The reduced knowledge base infers the final conclusions directly from the case inputs without any intermediate conclusions. The RTLS looks for the right input–output mapping without checking the validity of the retranslated intermediate rules by any domain expert. Despite the fact that RTLS generates correct final outputs for the validated cases, the outcome does not guarantee that the expert system yields the correct conclusions because there is no accurate retranslation. The retranslation is necessary, because the RTLS output is a reduced rule base, which is difficult to interpret by humans as there are no intermediate conclusions. Thus, it cannot be assumed that the retranslated RTLS rule base yields always the same solutions as the refined rules found by RTLS[2]. As far as known by the authors, there is no *exact* retranslation approach for reduced rule bases yet (Boswell 1999; Knauf et al. 2002a). Surprisingly, GINSBERG stated that his RTLS approach is (only) suited for medium size rule bases: "For large scale problems it will undoubtedly be necessary to employ heuristic strategies in order to pinpoint selected portions of the theory for reduction or partial reduction"[3].

GINSBERG's work didn't find a complete agreement: "Although the retranslation algorithm presented in (Ginsberg 1990) has been tested on a small size medical expert system . . . , there are several open questions about it. For example, are the new rules introduced as a result of the retranslation process acceptable from the semantic point of view? Can they introduce new structural anomalies in the knowledge base (for example, redundancies)?"[4]. Based on this insight, ZLATAREVA developed the retranslation–free revision system VVR (Zlatareva 1994).

There are two other refinement approaches that should be mentioned in this context: (1) KRUSTWORKS/KRUSTTOOL, and (2) STALKER. KRUSTWORKS targets the development of generic refinement operators that can be assembled individually so that the toolkit KRUSTTOOL can cope with a certain refinement problem (Boswell and Craw 2000). A KRUSTTOOL performs a three stage process: (1) blame allocation (to identify faulty rules), (2) refinement creation, and (3) rule refinement selection. It creates many alternative refinements, but employs hill climbing procedures for the selection of the best rule refinements. Several systems employ a so–called *radicality ordering* with respect to the sequence among the available refinement operators. For example, a refinement operation *component addition* cannot be executed before an on–target refinement operation *component deletion* took place (Boswell 1999). This hill climbing approach is typical for the current state of the art. There is no satisfactory methodical standard concerning the selection of rule refinements. The distinctive feature of the refinement system STALKER is a Truth Maintenance System ($TMS$) for speeding up the refinement and testing. For medium size rule bases, it has been shown that STALKER (Carbonara and Sleeman 1999) is about 50 times faster than KRUST.

In (Knauf 2000), a new case–based reduction approach has been presented in the context of a novel validation methodology. A discussion of this reduction approach in the context of the ones mentioned above is the focus of the following sections. A characteristic of this approach is the case associated competence estimation for each validator which is utilized to judge the system's validity. This is because the experts' knowledge is also validated.

The current multiple case analysis systems are executing input–output validation only. Unlike rule trace validation, the input–output validation is not facing the validity of intermediate inference results. Actually, there is no multiple case analysis system that faces the validity of the reasoning traces, in particular the intermediate conclusions. Furthermore, there is no mathematical optimization for the selection of the best rule refinements. Currently, hill climbing methods are used for this purpose. In (Ginsberg 1988c), the author states that mathematical optimization cannot be applied for the optimal selection of rule refinements. Fortunately, this is not true, since (Kelbassa 2003) introduces an operations research approach to the optimal selection of alternative and competing rule refinements.

However, as upcoming results the authors expect the development of a generic validation interface for rule retranslation. This could be managed by remote validation via the Internet. That increases the chance to recruit a competent international topical expert panel.

## The Retranslation Problem

### GINSBERG's Retranslation Approach

GINSBERG's approach performs rule retranslation in a relaxed manner. The basic idea of this approach is to start with the reasoning endpoints (final conclusions) and to retranslate first the maximal level $n$, and then the levels $n-1$ down to the input level $0$ (Ginsberg 1990). This retranslation step uses the rules of the unreduced *old* rule base and looks for modifications that are compatible with the revised theory. If the rules for the endpoint level have been retranslated, then the *back–propagating* succeeds. Important for retranslation are several relations as *eigen–terms*, i.e. hypotheses, which support one final conclusion only, and so–called *rule corre-*

[1] Dieses Paper hat leider Druckfehler auf S. 281, Zeilen 3 und 7: Richtig ist $\Delta I := \{I_0^k \Delta I_1^k\}$ und $\Delta R^k := \{R_0^k \Delta R_1^k\}$, wobei $\Delta$ die symmertische Differenz ist.

[2] Ginsberg 1990, p. 782: "This means that the new theory generated by this method may, and generally will, correspond to a reduction that is not identical to the input from RTLS."

[3] Ginsberg 1988b, p. 595

[4] Zlatareva and Preece 1994, p. 159

*lated theoretical terms*, and *theory correlated observables*. If there are two cases, which have the same observables at the input level, but different final conclusions, then GINSBERG's approach is looking for other observables in order to come up with rules that distinguish both final outputs in the new rule base.

However, the relaxed retranslation is lacking trace validation. In GINSBERG's approach there is no final validation of the obtained reasoning path despite it is being known that the retranslated rule base does not yield the same performance as the revised reduced one does. In principle, the processing of relations like *eigen–terms*[5] and theory–correlated theoretical terms cannot ensure that the generated conclusions are semantically valid with respect to the cases that are subject to examination respectively validation. Therefore, the authors propose a validation step, which examines the multi–level reasoning trace after rule refinement is performed.

## KNAUF's Retranslation Approach

KNAUF's retranslation approach is a part of a test case–based methodology for the validation of rule based systems (Knauf 2000; Knauf et al. 2002b). The developed technology covers five steps: (1) test case generation, (2) test case experimentation, (3) evaluation, (4) validity assessment, and (5) system refinement. The validity assessment leads to different validity degrees, which are associated with outputs, rules, and test data. Based on these validities, the last step leads to a new, restructured rule base that maps the test case set exactly to the solution that obtained the best rating from the expert panel in the validation session.

The system refinement based on *better knowledge* provided by experts has to be considered in the context of *learning by examples*. There are plenty of formal learning approaches that solve tasks like this. Usually, they aim at developing rules that map test data with *known* solutions to their correct solution.[6] Unfortunately, these formal methods lead to rules that might reflect reality fairly well, but are not *readable* or *interpretable* by domain experts. Even worse, these refinement systems might construct rules that reflect the cases (examples) correctly, but are wrong with respect to the causal connection they express.

KNAUF's refinement idea is to come up with a formal rule reconstruction procedure based on the validation results: validity degrees associated with test cases, rules, and outputs. The result of the refinement is a reduced rule base, which consists of *one–shoot rules*, which infer directly the right output from the input given by the considered cases without intermediate conclusions (hypotheses). Therefore, this reduced rule base must be retranslated to make the rules easier to understand and to interpret by human experts because of their shorter *if*–parts. For retranslation, pre–compiled knowledge is utilized, which occurs as rules with an intermediate conclusion (hypothesis) as their *then*–parts.

[5] If there is significant scientific progress in the domain under evaluation, then eigen–terms may become useless or obsolete.

[6] This property is usually called consistency.

## Validation of Rule Retranslations

In the previous sections the functions that should be performed by a generic rule validation interface have been introduced. For a summary and overview, they are listed here:

- Approval or rejection of final rule–based system outputs (Ginsberg 1988c; Knauf 2000; Knauf et al. 2002b).
- Evaluation of all conclusions, i.e. all intermediate and final ones (Kelbassa 1990, 2003).
- Acquisition of validation knowledge from human domain experts by validation technologies like (Knauf 2000; Knauf et al. 2002b).
- Acquisition of reasoning faults and target rule trace knowledge (Kelbassa 1990, 2002, 2003).
- Acceptance or rejection of rule revisions suggested by the validation system (Ginsberg 1988c; Davis and Lenat 1982; Kelbassa 1990; Boswell 1999).
- Detach the identification of invalidities from the final determination of the effective refinements (Kelbassa 1990).
- Acceptance or rejection of rule refinements suggested by the rule retranslation system.

It turned out that coping with the rule retranslation problem yields the need for evaluation of alternative rule retranslations. Similar to the TURING Test - like validation interface (Knauf 2000) the validation system should provide a generic interface for the evaluation of the retranslated rules in the generated rule trace. Thus, we need two different validation modes (1) a multi–level reasoning evaluation mode and (2) a single rule evaluation mode. The *multi–level reasoning evaluation mode* presents the case–related reasoning trace with intermediate and final conclusions. From the input level, it leads via intermediate conclusions to the final outputs.

Figure 1 shows a generic multi–level validation interface for intermediate and final conclusions. The conclusions are *mouse–sensitive* so that the validator can mark a certain intermediate or final conclusion that is not valid in his eyes. In the example of figure 1, the hypotheses $h_{12}$, $h_{122}$, and the output $o_{17}$ are marked as invalid. Based on these marks provided by the validator, the validation system is able to identify the invalid rules.

| |
|---|
| If $\bullet A$ $\circ$AND$\bullet$NOT $B$<br>Then $\bullet Z$ |

refined into $\hookrightarrow$

| |
|---|
| If $\bullet A$ $\bullet$OR $\bullet$NOT $B$<br>Then $\bullet Z$ |

Figure 2: Single rule validation example

Figure 2 shows an example for the *single rule evaluation mode*. Here, the conditions and the single final conclusion are presented to the validator who can change a *validity flag* $\bullet$ (valid) respectively $\circ$ (invalid) by mouse–click, so that invalid conditions are revealed. As shown above, the multi–level validation interface offers a top–down approach: first, invalid conclusions are marked and second, the particular *guilty* rule can be inspected by the validator.

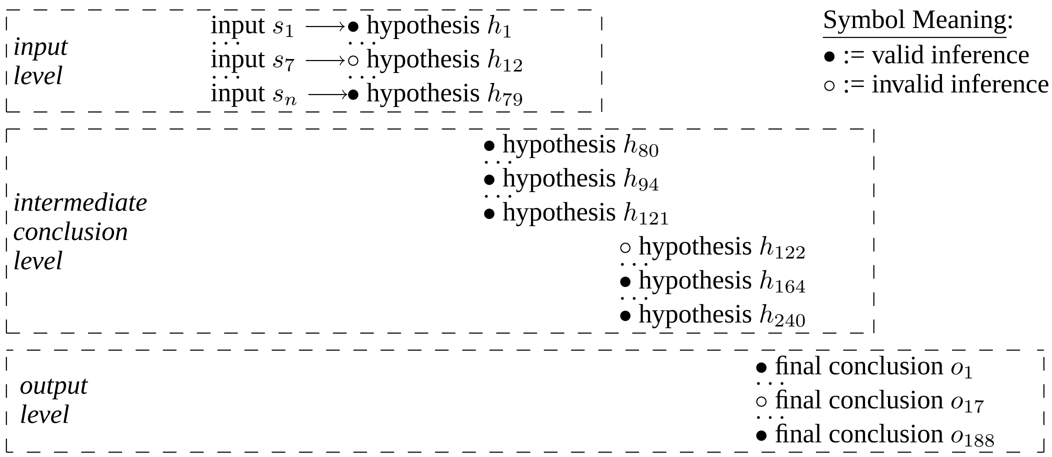


Figure 1: Generic validation interface for intermediate and final conclusions

## The two Stage Validation Process

The discussion so far provided an overview on the state of the art of rule validation and refinement. Here, we introduce a two stage validation process, which combines the approaches of KNAUF (Knauf 2000; Knauf et al. 2002b) and KELBASSA (Kelbassa 1990; Kelbassa 2003) and enjoys the advantages of both. The process suggested here covers KNAUF's approach in a first stage that is continued by KELBASSA's approach as the second stage. The objective of this combination is to perform a reasoning trace validation with the best available validators as illustrated in figure 3.

The first stage of the proposed combined validation process realizes KNAUF's approach and yields three main results: (1) a case–related competence estimation for the involved validators (2) the valid input–output mapping[7], and (3) the revised rule base (after a relaxed retranslation step). As we are able to identify the validator(s), who is (are) most competent for a test case, the second stage can be performed by presenting each case to the most competent validator(s).

However, this revised rule base $RB^{1*}$ is just utilizing pre–defined knowledge, which contributes to the valid input–output behavior but which is not validated itself. We call it a *relaxed result* concerning the reasoning trace and the intermediate inferences. Therefore, the second validation stage is focussing the validity of the reasoning path for all cases under examination. The validation expertise obtained in the second validation stage is the subject of mathematical optimization as described in (Kelbassa 2003). Thus, the overall result is a refined rule base $RB^{2*}$ obtained by the optimal selection of the best rule refinements.

## Conclusion

Formal refinement approaches like the one in (Knauf 2000; Knauf et al. 2002b) aim at the total validity of the input–output behavior of the rule–based expert system. A little (formal, not topical) retranslation is performed by utilizing the intermediate conclusions that have been a part of the knowledge base before refinement. These intermediate conclusions itself are not validated and might be wrong. Topical refinement processes like the one in (Kelbassa 2003), on the other hand, aim at the validation of the inference path and optimal rule base refinement.

Here, we proposed a useful combination of both: A two stage validation interface. In a first stage KNAUF's approach is performed. One result of applying this approach is a case–associated competence estimation of each involved validator. This is a useful basis to apply KELBASSA's approach that forms the second stage. By utilizing the competence estimation this second stage is performed within an interactive dialogue with the most competent human expert(s) for the considered (sub–) domain of expertise.

On a first view there seems to be a remaining problem: A rule might be used to infer different input–output cases that have different *most competent* experts. KNAUF's refinement aims at inferring an *optimal solution* for each case. If a rule produces different optimal solutions provided by different experts for different inputs, this rule is *split* into several rules that have the same *best expert* each. This rather accidental feature of KNAUF's refinement idea became important for the compatibility with KELBASSA's approach.

[7] Valid, in this context means the mapping that met the maximal experts' approval

## References

Boswell, R. 1999. Knowledge Refinement for a Formulation System. Ph.D. Thesis, Dept. for Computing and Mathematical Science, The Robert Gordon University, Aberdeen / Scotland, U.K.

Boswell, R.; Craw, S. 2000. Experiences with a Generic Refinement Toolkit. *Proc. of the Europ. Knowledge Acquisition Workshop 2000 (EKAW 2000)*, pp. 249–256.

Carbonara, L.; Sleeman, D. 1999. Effective and Efficient Knowledge Base Refinement. *Machine Learning*, 37, pp. 143–181.

Collet, M.; Hälker–Küsters, M.; Tischendorf, M. 1990. Ergebnisse einer Untersuchung über den Einsatz von Ex-

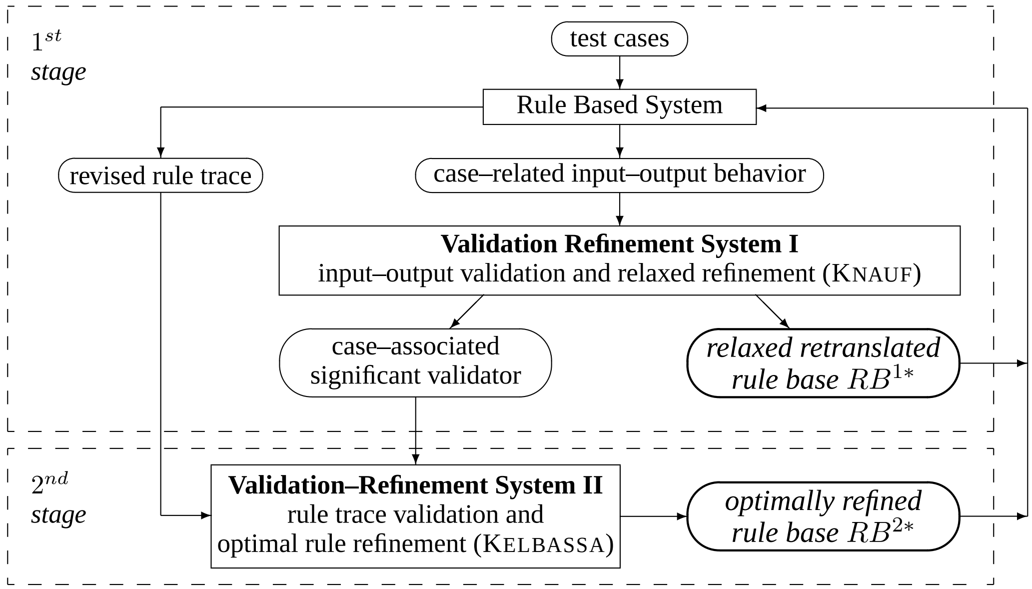


Figure 3: The two stage validation process for rule bases

pertensystemen im Bereich Umwelt. Pillmann / Jaeschke (eds): *Informatik für den Umweltschutz* Informatik–Fachberichte 256. Berlin: Springer, pp. 167–176.

Craw, S. 1991. Judging Knowledge Base Quality. M. Ayel / J.P. Laurent (eds): *Validation, Verification and Test of Knowledge–Based Systems*, N.Y.: John Wiley & Sons, pp. 207–219.

Davis, R.; Lenat, D.B. 1982. *Knowledge–Based Systems in Artificial Intelligence.* New York: McGraw Hill Int. Book Company.

Ginsberg, A. 1988a. Knowledge Base Reduction: A New Approach to Checking Knowledge Bases for Inconsistency and Redundancy. *Proc. of the $7^{th}$ Annual Conf. on AI (AAAI–88)*, pp. 585–589.

Ginsberg, A. 1988b. Theory Revision via Prior Operationalization. *Proc. of the $7^{th}$ Annual Conf. on AI (AAAI–88)*, pp. 590–595.

Ginsberg, A. 1988c. *Automatic Refinement of Expert System Knowledge Bases.* London: Pitman Publishing.

Ginsberg, A. 1990. Theory Reduction, Theory Revision, and Retranslation. *Proc. of the $9^{th}$ Nat. Conf. on AI (AAAI–90)*, Boston, Mass., pp. 777–782.

Kahn, G.; Nowlan, S.; McDermott, J. 1985. MORE: An Intelligent Knowledge Acquisition Tool. *Proc. of the $9^{th}$ Int. Joint Conf. on AI (IJCAI–85)*, Los Angeles, California, vol. 1, pp. 581–584.

Kelbassa, H.–W. 1990. Fallbezogene Revision und Validierung von regelbasiertem Expertenwissen für die Altlastenbeurteilung. Pillmann/Jaeschke (eds): *Informatik für den Umweltschutz*, Berlin: Springer, pp. 276–285.

Kelbassa, H.–W. 2002. Context Refinement - Investigating the Rule Refinement Completeness of SEEK/SEEK2. *Proc. of $15^{th}$ Europ. Conf. on AI 2002 (ECAI 2002)*, Lyon, France, pp. 205–209.

Kelbassa, H.–W. 2003. Selection of Optimal Rule Refinements. *Proc. of $16^{th}$ Int. Florida AI Research Society Conf. 2003 (FLAIRS-03)*, St. Augustine, Florida, USA, Menlo Park, California: AAAI Press.

Knauf, R. 2000. *Validating Rule–Based Systems: A Complete Methodology.* Habilitation Thesis, ISBN 3–8265–8293–4, Aachen: Shaker.

Knauf, R.; Philippow, I.; Gonzalez, A.J.; Jantke, K.P.; Salecker, D. 2002a. System Refinement in Practice – Using a Formal Method to Modify Real Life Knowledge. *Proc. of $15^{th}$ Int. Florida AI Research Society Conf. 2002 (FLAIRS-02)*, Pensacola Beach, Florida, USA, pp. 216–220, Menlo Park, California: AAAI Press.

Knauf, R.; Gonzalez, A.J.; Abel, T. 2002b. A Framework for Validation of Rule–Based Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 32, # 3, pp. 281–295.

Marcus, S. (ed) 1988. *Automating Knowledge Aquisition for Expert Systems.* Boston: Kluwer Academic Publishers.

Murphy, R.M.; Pazzani, M.J. 1994. Revision of Production System Rule Bases. *Proc. of the $11^{th}$ Int. Conference on Machine Learning*, pp. 199–207, New Brunswick, New Jersey: Morgan Kaufman.

Zlatareva, N; Preece, A. 1994. State of the Art in Automated Validation of Knowledge–Based Systems. *Expert Systems with Applications*, vol. 7, # 2, pp. 151–167.

Zlatareva, N. 1994. A framework for verification, validation, and refinement of knowledge bases: The VVR system. *Int. Journal of Intelligent Systems*, 9(8):703-738.