Knauf, Rainer; Gonzalez, Avelino J.; Jantke, Klaus P. :

Validating rule-based systems: a complete methodology

Zuerst erschienen in:
IEEE SMC '99 conference proceedings / 1999 IEEE International Conference on Systems, Man, and Cybernetics : October 12- 15 1999, Tokyo. - Piscataway, NJ : IEEE Service Center, 1999, Vol. 5, S. 744-749
DOI: 10.1109/ICSMC.1999.815644

Validating Rule-Based Systems: A Complete Methodology

Rainer Knauf* & Avelino J. Gonzalez** & Klaus P. Jantke***

* Technical University of Ilmenau, Faculty of Computer Science and Automation
PO Box 10 05 65, 98684 Ilmenau, Germany
Rainer.Knauf@theoinf.tu-ilmenau.de

** University of Central Florida, Dept. of Electrical and Computer Engineering
Orlando, FL 32816-2450, USA
ajg@ece.engr.ucf.edu

*** German Research Institute of Artificial Intelligence Ltd.
Stuhlsatzenhausweg 3, 66125 Saarbrücken, Germany
jantke@dfki.de

ABSTRACT

This paper describes a complete methodology for the validation of rule-based expert systems. The methodology is presented as a 5-step process that has three central themes: (1) creation of a minimal set of test inputs that adequately cover the domain represented in the knowledge base, (2) a Turing Test-like methodolgy that evaluates the system's responses to the test inputs and compares them to the responses of human experts, and (3) use the validation results for system improvement.

The development of minimal set of test inputs takes into consideration various criteria, both user-defined and domain-specific. These criteria are used to reduce the potentially very large exhaustive set of test inputs to one that is practical.

The Turing Test-like evaluation methodology makes use of a panel of experts to both evaluate each set of test cases and compare the results with those of the expert system, as well as with those of the other experts in the validation panel.

The hypothesis being presented here is that much can be learned about the experts themselves by having them evaluate each other's responses to the same test inputs anonymously. Thus, by carefully scrutinizing the results of each expert in relation to the other experts, we are better able to judge an evaluator's expertise, and consequently, better determine the validity of an expert system.

Another contribution presented here is a system refinement strategy based on (1) determining the rules, which are "guilty" of the system's invalidity, and (2) determining "better" solutions to those test cases, which obtained "bad marks" by the validation panel.

Lastly, the work describes a partial implementation of the test input minimalization process on a small but non-trivial expert system.

1. INTRODUCTION

There is abundant evidence of the need for an integrated approach towards validation and verification of complex systems. In [1], the authors clearly point out that "the inability to adequately evaluate systems may become the limiting factor in our ability to employ systems that our technology and knowledge will allow us to design".

Here, we follow the approach of O'Keefe and O'Leary ([2]) who characterize **verification** and **validation** as *building the system right*, and *building the right system*, respectively.

The **verification** provides a firm basis for the question of whether or not a system meets its specification. In contrast, **validation** asks whether or not a system is considered to be the required one, something that somehow lies in the eye of the beholder. The essential difference is illustrated in figure 2.

We concentrate on the validation portion, as that is the one more closely related to ensuring appropriate response to inputs. The heart of the presented methodology is a TURING Test - like technology of a systematic system interrogation, which is composed of the following related steps ([3]):

1. **Test case generation** Generate and optimize a set of test input combinations (test data) that will simulate the inputs to be seen by the system in actual operation.
2. **Test case experimentation** Since intelligent systems emulate human expertise, it is clear that human opinion needs to be considered when evaluating the correctness of the system's response.
3. **Evaluation** This step interprets the results of the experimentation step and determines errors attributed to the system and reports it in an informal way.

0-7803-5731-0/99/$10.00 ©1999 IEEE

4. **Validity assessment** This step analyzes the results reported above and reaches conclusions about the validity of the system.

5. **System refinement** In order to improve the final system, this step provides guidance on how to correct the errors detected in the system as a result of the previous 4 steps. This, hopefully, leads to an improved system.

These steps are iterative in nature, where the process can be conducted again after the improvements have been made. Figure 1 illustrates the steps outlined above.

2. FUNDAMENTALS

In the minimal formal setting assumed so far, there are only two sets I and O. On these sets, a target relation $\mathcal{R} \subseteq I \times O$ is given.

There are two requirements to characterize expertise: An expert's knowledge should be **consistent** with the target phenomenon, i.e. there should be no cotradiction between both, and **complete**, i.e. from the possibly large amount of correct answers to an admissible question, an expert should know at least one.

A certain expert's knowledge $\mathcal{E}_i$ about a target domain $\mathcal{R}$ is assumed to be a particular relation $\mathcal{E}_i \subseteq I \times O$ such that the following requirements of expertise are satisfied:

$$\mathcal{E}_i \subseteq \mathcal{R} \qquad \textbf{[Exp1]}$$

$$\pi_{inp}(\mathcal{E}_i) = \pi_{inp}(\mathcal{R}) \qquad \textbf{[Exp2]}$$

Ideally, an expert's knowledge contains exactly the target relation:

$$\mathcal{E}_i = \mathcal{R} \qquad \textbf{[Omn]}$$

[Exp1] is a condition of consistency, **[Exp2]** is a condition of completeness, and **[Omn]** is a property called **omniscience**. An expertise $\mathcal{E}_i$ is said to be competent, exactly if it is complete and consistent:

$$competence = consistency + completeness$$

Practically, because of not directly knowing $\mathcal{R}$, we estimate $\mathcal{R}$ by $\bigcup_{i=1}^{n} \mathcal{E}_i$.

Based on these formalisms, we are now able to develop our validation scenario:

- There is assumed a (non-accessible) desired target behavior $\mathcal{R} \subseteq I \times O$.
- There is a team of n experts which is considered to be omniscient as a team.
- There is a system to be validated with an input/output relation $\mathcal{S}$.

Our validation methodology deals with relating the system's behavior to the experts' knowledge. Figure 2 is incorporating the formal concepts. A deeper discussion of the fundamentals can be found in [4], e.g.

3. GENERATION OF TEST CASES

One standard that does exist, however impractical as it may be in most cases, is the exhaustive testing of the system. For systems which have more than a few inputs, the combinations of values of these inputs can be prohibitively large, thus making exhaustive testing quite impractical. Nevertheless, it is not necessary in most cases to have a truly exhaustive set of test cases, and yet still be able to test the system in a *functionally exhaustive* fashion.

A *functionally exhaustive* set of test cases can be made considerably smaller than a *naively exhaustive* set by eliminating functionally equivalent input values and combinations of input values which subsume other values. Nevertheless, even this functionally exhaustive set is usually too large for practical purposes. Thus, there is a need for further reduction. Of course, one has to pay for it with a loss of functional exhaustivity.

A reasonable way to reduce the functional exhaustive set of test cases is to use *validation criteria*, which can be domain-, input-, output-, expert-, validator- or user-related in nature. These criteria are useful in determining a *test sufficiency level* for each test case of the functional exhaustive set. This test sufficiency level can be used as an indicator for the decision whether or not a given test case is really needed from a practical standpoint.

Thus, we developed an automated technique which generates a functionally exhaustive set of test cases, and reduces it to a "reasonable" set of test cases by using validation criteria.

Due to simplification reasons but also because of its practical relevance, we consider rule-based systems with an input I of an m-dimensional "input space", in which each dimension is "atomic", i.e. not compound in any way, and an output O of a set of possible output values.

The main test case generation idea is

1. to generate a "quasi exhaustive" set of test cases ($QuEST$) (cf. [5]), and
2. to define some validation criteria and to use them for a reduction of $QuEST$ down to a "reasonable" set of test cases ($ReST$) as described in [6].

The generation procedure contains a step of analysing the dependencies between the inputs and outputs of the system, which is a basis for the reduction procedure, which needs the so called *dependency sets*, which

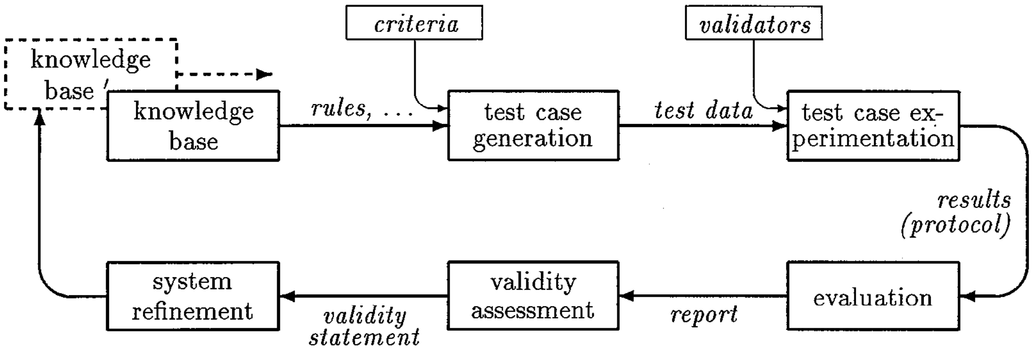

Figure 1: Steps in the Proposed Validation Process

describe which output depends on which inputs[1] and a set of so called *critical values*, which describe certain values of a single input which are considered a trigger value for the output[2].

4. A TURING TEST EVALUATION

Figure 2 sketches the scenario of validation, which is commented here. We are convinced, that there always remain some gaps between the (non-formalized) real domain knowledge, and the formalized knowledge of an AI system:

1. The first gap is the one between the desired target domain behavior $\mathcal{R}$ and the experts' knowledge $\mathcal{E}_1, \ldots, \mathcal{E}_n$. This is the gap marked by grey arrows in figure 2.

2. The second gap is between the experts' knowledge $\mathcal{E}_1, \ldots, \mathcal{E}_n$ and the system's specification, which is (in case of successful verification) equivalent to $\mathcal{S}$.

Unfortunately, earthly creatures like humans are not cabable of bridging the first gap. A technology to bridge the second gap is the subject of this section.

The suggested methodology is quite similar in concept to the TURING test. It uses

- *one* AI system which is to be validated,
- n experts, and
- a "reasonable" set of m test cases *ReST* respectively its input-part, the test data.

The idea of the TURING test methodology, as illustrated in figure 3, is divided into four steps:

1. solving of the test cases by the expert validation panel as well as by the expert system to be validated,
2. randomly mixing the test case solutions and removing their authorship,
3. rating all (anonymous) test case solutions, and
4. evaluating the ratings.

Since intelligent systems emulate human expertise, it is clear that human opinion needs to be considered when evaluating the correctness of the system's response. But human experts can vary in their competence, their own self-image, and their bias for or against automation. Furthermore, competence is usually not distributed "homogeniously" in the entire "input space" of the system.

Thus, it is important that an efficient and objective method exist to fairly evaluate the correctness of the system's outputs given imperfect and inhomogeniously distributed human expertise.

That's why the experimentation session consists of exercising the resulting set of test data by the intelligent system as well as by the one or more validating experts in order to obtain and document the responses to each test data by the various sources.

These responses will be presented to the experts anonymously, i.e. the rating expert does not know, which source a present solution came from. Among the presented solutions there is the system's solution and the own one.

Besides the chance to express some incompetence directly in the test case solution session, the experts can express a lack of competence indirectly in the rating session by giving the own solution bad marks and/or admitting that they are not sure with the rating of some solutions.

[1] In most practical cases even non-experts can exclude some kinds of dependencies: in medical diagnosis there is no dependency between the patient's temperature and the gastric ulcer disease, in technical diagnosis of cars there is no dependency between a non-starting motor and tire pressure, e.g.

[2] Once more: in most practical cases such values can be found even without having some deeper background knowledge about the domain: Trigger values of this kind are 38 degrees centigrade of a patient's temperature or a certain gas consumption of a car, e.g.

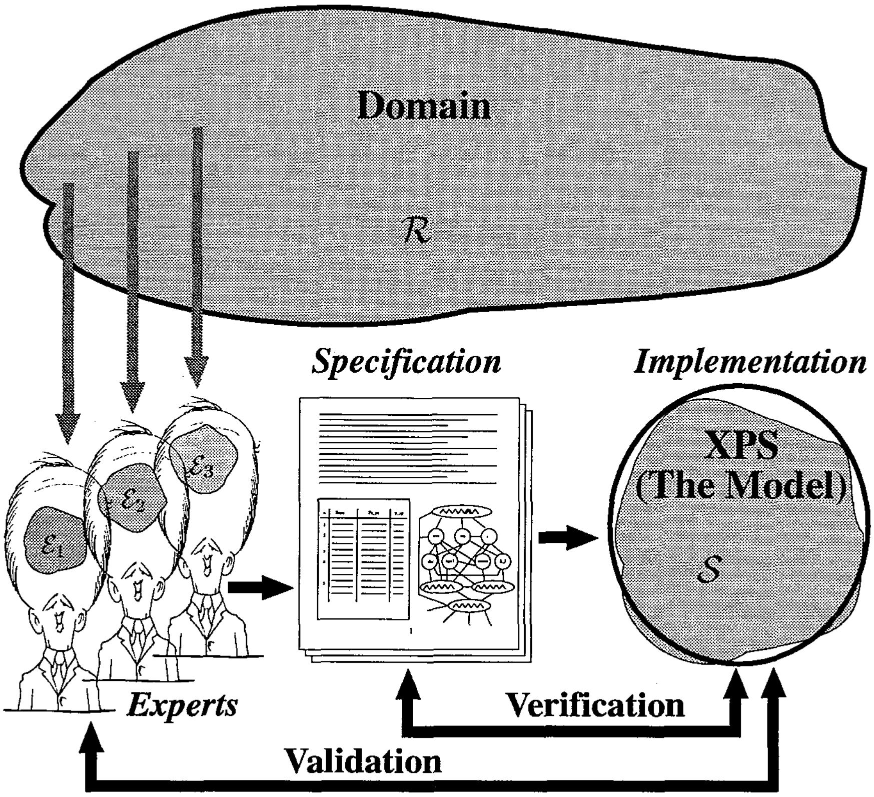

Figure 2: Relating Validation and Verification Based on a Minimal Setting of Formalized Human Expertise

To come up with a validity assessment for the system we consider the expert's assesments of the system solution, but each assessment is weighted with a "local competence" of the rating expert for the considered test case.

This "local competence" of an expert e_i for a test case t_j is estimated by considering

- the expert's behavior while solving the test case t_j (Did he/she admit some incompetence by giving the solution *"I don't know!"*?),
- the expert's behavior while rating the test case solutions (Did he/she give his/her own solution bad marks after knowing the colleagues' soltions?, How often did he/she express certainty while rating the solutions for t_j ?), and
- the other experts' assessment of the solution of the expert e_i.

A deeper discussion of these steps and the competence assessment can be found in [7], e.g.

The TURING test methodology leads to both a validity assessment for use in system evaluation by humans and a validity assessment for use in system refinement.

5. SYSTEM REFINEMENT

The approach of the previous section delivers a local validity v_i for each test case $[t_j, sol_j]$ as one of the results.

For a rule-based system with a rule set R, there is a way to determine the set of rules $R_j \subseteq R$, which contribute to a considered test case $[t_j, sol_j]$.

Of course, it may happen that one rule has been used for a considered test case more than once.[3] Consequently, we should allow a multiple occurence of the same rule in R_j. Thus, R_j is a multiset.

Based on the knowledge, how the rules contribute to each test case and the validity of each test case we can estimate the validity of each rule (cf. [8], e.g.) and thus, point out the weaknesses of the systems in particular.

Actually, we are developing a technology to come up with

- "hypothesis-associated" validities of the system, and
- validities of the alternative solutions delivered by the experts during the experimentation session.

[3] Especially for planning and configurating systems this is conceivable.

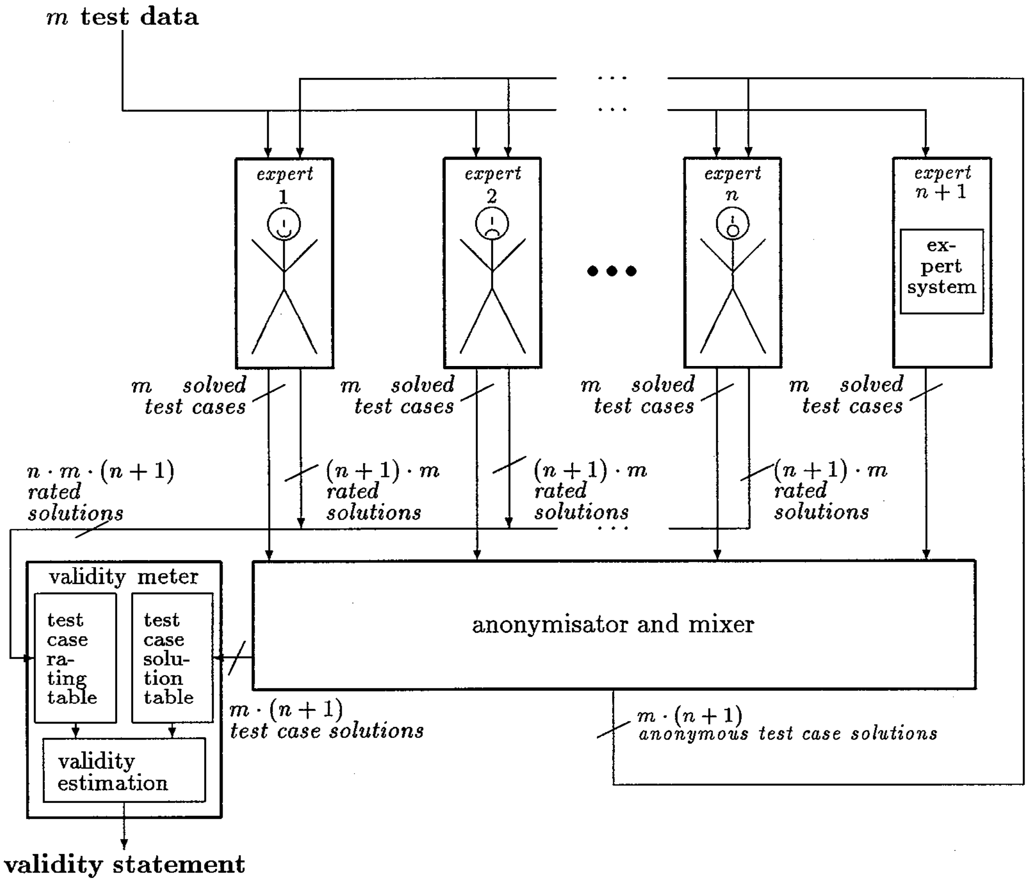

Figure 3: A survey of the TURING test to estimate an AI System's validity

For the latter we use the same technology to estimate the validity of an expert's solution as we used for the assessment of the system's soltuion.

After having both our research will be focused on creating new rules, which deliver the highest rated solution for each test case out of *ReST*.

Unfortunately, this is still a subject of research but we are optimistic to close the "validation loop" of figure 1 pretty soon.

6. EVALUATION OF THE METHODOLOGY

Of course, we tried to validate our validation technology as far as we can. We put into practice the concept of the *quasi-exhaustive set of test cases QuEST* and empirically determined its validity by applying it to a small but non-trivial expert system.

The basic idea behind this work was to determine whether errors artificially seeded into the selected test bed expert system would be detected by the test cases included in the *QuEST*.

As a test bed we used a small rule-based classification system. Its purpose is to identify birds that can bc observed in Florida. Its knowledge base consists of 71 rules which are processed in a backward-chaining way.

The original system was treated as the "expert" whose answers were always correct. Additionally, errors were seeded into copies of the original system to see whether the test cases that composed the *QuEST* and the (finally reduced) *ReST* were able to detect them.

The test data were presented both to the faulty system as well as to the original (the "expert"). Differences in their responses were indicative of an error in the faulty version and of success in detecting that error through the *QuEST*.

The system was modified 36 times, each version introducing one and only one error. Of the 36 errors introduced, 10 went undetected. However, a closer examination of the cause of the undetected errors provided a valuable insight into the workings of expert systems.

9 of the 10 undetected errors could be easily detected by making some rather simple adjustments to the *QuEST* generation procedure. The last one represented an error of omission, and therefore not detectable.[4]

[4] Detecting this kind of error is rather an objective of verifi-

36 cases do not represent a large number of errors in order to definitely declare success or failure. However, these cases represent the different types of errors that could exist in an knowledge-based system. Therefore, the authors consider the *QuEST* to be in fact functionally equivalent to the (really) exhaustive set of cases, if the recommendations included herein are implemented. Further research in this topic, of course, centers on the *ReST*, and how to reduce that number.

7. SUMMARY

The main difficulty in validation of AI systems is the fact that the target domain is neither directly accessible nor is there a commonly accepted formal description of it. Thus, the only chance to validate these systems is to confront the system with representative test cases and a comparison of the system's answers with the answers of humans, who are considered to be experts in the application field.

The heart of the presented methodology is a TURING test-like technique of a systematic interrogation of the system through test data.

The present paper outlines some ideas on how to ensure, that

1. the coverage of the "input space" is both, as complete as possible and as small, but as representative as possible,
2. the selected test cases are used in a practical experimentation,
3. the system's responses are evaluated as objectively as possible to be correct or not correct,
4. this evaluation can be used to create a validity assessment that is able to be interpreted by humans, and
5. there is also a validity assessment, which is useful for a system's refinement.

These ideas refer to rule-based systems, which is the most commonly used kind of system in real world applications.

Altough the present paper promises an entire and complete framework for validation of rule-based systems, it could not meet this expectations completely, of course. Besides the need of some remaining research in this field the authors feel, that there is an urgent need for concepts of validation for other kinds of systems as well, and of "intelligent behavior" in general.

cation than a one of validation.

7. REFERENCES

[1] J.A. Wise, M.A. Wise, "Basic considerations in verification and validation", Wise/Hopkin/Stager(eds): *Verification and Validation of Complex Systems: Human Factors Issues.* vol. 110 of NATO ASI Series F: *Computer and System Sciences*, Springer-Verlag, 1993, pp. 87-95

[2] R.M. O'Keefe, D.E. O'Leary, "Expert system verification and validation: A survey and tutorial", *Artificial Intelligence Review*, vol. 7, 1993, pp. 3-42

[3] R. Knauf, Th. Abel, K.P. Jantke, A.J. Gonzalez, "A framework for validation of knowledge-based systems", Grieser/Beick/Jantke(eds): *Proc. Int. Workshop on Aspects of Intelligent Systems Validation*, Ilmenau, Germany, January 1998, Technical Report MEME-MMM-98-2, Hokkaido University, Meme Media Laboratory, Sapporo, Japan, April 1998, pp. 1-19

[4] K.P. Jantke, Th. Abel, R. Knauf, "Fundamentals of a turing test approach to validation" Hokkaido University, Meme Media Laboratory, *Intelligent Meme Report MEME-IMP-1/1997*, Sapporo, Japan, 1997

[5] Th. Abel, R. Knauf, A.J. Gonzalez, "Generation of a minimal set of test cases that is functionally equivalent to an exhaustive set, for use in knowledge-based system validation", Stewman(ed): *Proc. of the 9th International Florida Artificial Intelligence Research Symposium (FLAIRS-96)*, Key West, FL, USA, May 20-22, 1996, pp. 280-284

[6] Th. Abel, A.J. Gonzalez, "Utilizing criteria to reduce a set of test cases for expert system validation" Dankel(ed): *Proc. of the 10th International Florida Artificial Intelligence Research Symposium (FLAIRS-97)*, Daytona Beach, FL, USA, May 11-14, 1997, pp. 402-406

[7] K.P. Jantke, R. Knauf, Th. Abel, "The Turing test approach to validation" Terano(ed): *Proc. Workshop on Validation, Verification & Refinement of AI Systems and Subsystems (W32)*, Int. Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya (Japan), August 23-29, 1997; pp. 35-45

[8] R. Knauf, A.J. Gonzalez, "Validation of rule-based and case-based systems: What's common and what should be different" Beick/Jantke(eds): *Proc. of an International Workshop on Future Trends in Intelligent system Validation.*, Sapporo, Japan, Sept. 1998, Technical Report MEME-MMM-98-5, Hokkaido University, Meme Media Laboratory, pp. 25-40

[9] K.P. Jantke, R. Knauf, A. Stephan, "Validation von Anwendungssoftware: Von der Forschung zum Marktfaktor" Jantke/Grieser(eds): *Proc. of 5. Leipziger Informatik-Tage 1997 (LIT-97)*, Leipzig, Sept. 26, 1997, eingeladener Hauptvortrag, pp. 1-21