The Power of Experience

On the Usefulness of Validation Knowledge

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

Setsuo Tsuruta and **Kenichi Uehara**
School of Information Environment
Tokyo Denki Unversity
2-1200 Musai-gakuendai, Inzai
Chiba, 270-1382, Japan

Takashi Onoyama
Research and Development Department
Hitachi Software Engineering Co., Ltd.
6–81, Onoe-cho, Naka–ku, Yokohama, 231–8475, Japan

Torsten Kurbad
Knowledge Media Research Center
Konrad-Adenauer-Str. 40
72072 Tübingen, Germany

Abstract

TURING Test technologies are promising ways to validate AI systems which may have no alternative way to indicate validity. Human experts (validators) are often too expensive to involve. Furthermore, they often have different opinions from each other and from themselves over time. One way out of this situation is to employ a **Validation Knowledge Base** (VKB) which can be considered to be the collective experience of human expert panels. VKB is constructed and maintained across various validation sessions. Primary benefits are (1) decreasing the validators' workload and (2) refining the methodology itself. Additionally, there are some side effects that (1) improve the selection of an appropriate expert panel and (2) improve the identification of an optimal solution to a test cases. Furthermore, **Validation Experts Software Agents** ($VESA$) are introduced as an model of a particular expert's knowledge. $VESA$ is a software agent corresponding to a human validator. It systematically models the validation knowledge and behavior of its human origin. After a learning period, it can be used to substitute the human exert.

Introduction

Because of the character of their typical application fields, intelligent systems are validated and refined on the basis of human expertise. Experts have different beliefs, experiences, learning capabilities and are not free of mistakes. Their opinions about the desired system's behavior differ from each other and change over time. Their opinions differ from their previous ones, even in the same context, as a result of misinterpretations, mistakes or new insights. Furthermore, experts are too busy and too expensive to spend that much time for system validation and adjustment. Thus, the experts' workload for system validation is a serious issue.

To make validation results less dependent on the experts' opinions and to decrease the workload of the experts, the importance of storing and using historical validation results / knowledge in a Validation Knowledge Base (VKB) was originally proposed in (Tsuruta et al. 2000b) and adopted for a TURING Test validation technology in (Knauf, Gonzalez, and Tsuruta 2003).

In the technique described in (Knauf, Gonzalez, and Abel 2002), the result is influenced by the quality of interaction with human experts. Their excessive involvement is both time consuming and costly. In addition, human experts may not always be available or even willing to cooperate, thereby causing delays. In (Tsuruta et.al. 2002) this is summarized as *"The bottleneck in acquiring validation knowledge from experts who are busy."*

The validation procedure, as developed so far, covers five steps: (1) test case generation, (2) test case experimentation, (3) evaluation of results, (4) validity assessment, and (5) system refinement. These steps can be performed iteratively. Its most expensive part is the test case experimentation, because the test cases have to be solved and rated by both the system under examination and the humans who perform the examination.[1] This step is especially supported by the VKB. Furthermore, the VKB is applied to other useful purposes, for example

- to improve the validation methodology itself, e.g. to select experts for the validation panel, and
- to support the identification of an optimal solution among several candidate solutions.

Furthermore, a Validation Expert Software Agent ($VESA$) is developed based on the VKB. A $VESA$ keeps personal validation knowledge, such as previous validation judgments or the experiences of a human expert. It is an intelligent avatar corresponding to its human origin. At some point, a $VESA$ may be able to serve as a substitute for a missing human expert.

The following section describes the structure of the developed VKB. After this, the way to incorporate the VKB into the test case experimentation is outlined. The fourth

[1] In the process not only the system's solutions, but also the solutions provided by humans are examined. The latter is performed to estimate the experts' competence for each particular test case.

section is dedicated to the additional useful applications of the VKB. Since these effects can only be enjoyed by having a "well informed" VKB, the fifth section describes the way to gain as much validation knowledge as possible. The sixth section introduces the $VESA$ concept. Finally, all contributions are summarized and upcoming research directions are discussed.

The Contents of the VKB

The information which needs to be stored and maintained in the VKB for use in the *test case experimentation*, consists of the required input data, the produced output data, and some additional necessary data. According to the formal settings in (Knauf, Gonzalez, and Abel 2002) and (Kurbad 2003), the VKB contains a set of historical test cases, which can be described by 8–tuples

$$[t_j, E_K, E_I, sol_{K_j}^{opt}, r_{IjK}, c_{IjK}, \tau_S, D_C]$$

with

- t_j being a test case input,
- $sol_{K_j}^{opt}$ being a solution associated to t_j,
- E_K being a list of experts who provided this particular solution,
- E_I being a list of experts who rated this solution,
- r_{IjK} being the rating of this solution, which is provided by the experts in E_I,
- c_{IjK} being the certainty[2] of this rating,
- τ_S being a time stamp associated with the validation session in which the rating was provided, and
- D_C being an informal description of the application domain C that is helpful to explain similarities between different domains or fields of knowledge.

Additionally, a *list of supporters* $E_S \subseteq E_I$ for each solution $sol_{K_j}^{opt}$ is derived from this data. E_S is the list of rating experts who provided a positive rating for $sol_{K_j}^{opt}$.

The VKB is not completely transparent to all agents in the validation process. According to its purpose, some of the data is hidden to certain agents. For example, to ensure anonymity while solving and rating test cases within the TURING Test, E_K and E_I must not be presented to the expert panel of the current session. Furthermore, to ensure an unbiased rating, the historical rating r_{IjK} must not be presented to the expert panel that currently rates the solution.

Involvement of the VKB in the Test Case Generation and Experimentation

The intermediate results that occur during the experimentation as well as the VKB itself are stored in a relational database by using a *client–server database management system* ($DBMS$). This provides decentralized access to centralized data for clients, which work independently from each other. All data are kept central to the view of *knowledge engineer ($\sim$server)*, while only the necessary parts of it are shown to the *expert panel ($\sim$client)* (Kurbad 2003).

All experts in the panel take part in the experimentation session independently. By utilizing an HTML–based implementation approach for the client application, each expert is free to choose the time and place of his work. This effectively limits delays caused by experts who would otherwise be unavailable, as well as the cost of the validation process.

Figure 1 sketches the usage of VKB in the test case experimentation. After generating the so–called Quasi Exhaustive Set of Test Cases $QuEST$ (Knauf, Gonzalez, and Abel 2002), both $QuEST$ and the historical cases in VKB are subject to the criteria–based reduction procedure which aims at a subset of cases in $QuEST$ or VKB. This meets the requirements of the current application and is small enough to be the subject of the test case experimentation.

The VKB is a database of test cases and their associated solutions, which received an optimal rating in previous validation sessions. These solutions are considered an additional (external) source of expertise that does not explicitly appear in the solving session. Therefore, the cases originating from the VKB are not subject to the test case solving session.

Regardless of their former ratings, the cases from the VKB have to be rated by the current expert panel again for two reasons:

1. Topical domain knowledge of AI systems has a dynamic nature. It might have changed since the time, when the information in the VKB was acquired. This might be because of recent insights, but also because of modified application circumstances.
2. Additionally, there is a responsibility for the results of applying the validation technology, i.e. for the validity statements as well as for the refined knowledge base.

   These results need, when communicated and used for (commercial, political, ...) decisions, a clear association to responsible persons. Of course, the current panel that rated the solutions must serve as these responsible persons. Although there is already a (historical) rating for the test cases in the VKB, this panel must have the opportunity to provide its own ratings to these test cases.[3]

Fortunately, not all cases of the VKB that "survived" the criteria–based reduction process need to be rated again. Only cases which's solutions are different from the system's solution have to be involved in the rating process (see $sol = system's?$ – box in figure 1), because (1) we are only interested in new external knowledge that is outside the expertise of the expert panel and (2) the systems solution is in the process anyway[4] and the test case solving procedure

[2] Besides providing a rating that might be 0 (wrong) or 1 (correct), the experts have the opportunity to express, whether (c=1) or not (c=0) they feel certain while providing this rating.

[3] Nobody would agree to be responsible for something that he/she cannot control.

[4] The test case generation step exclusively produces test cases with system's solutions.

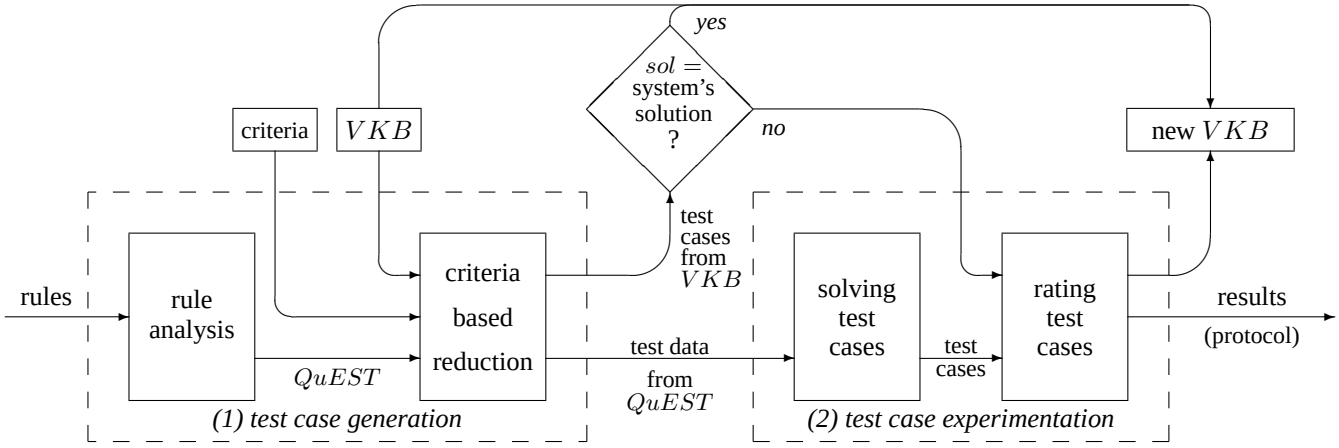

Figure 1: The use of the VKB in the Test Case Generation and Experimentation

additionally provides alternative ("man–made") solutions to it.

Utilizing the Experience of the VKB

As previously indicated, the knowledge gained in the VKB is also applied to other useful purposes:

1. It can be used for a refined competence estimation of the panel experts. This estimation is used as a weight an expert's rating of the system's solution to compute its validity degree (Knauf, Gonzalez, and Abel 2002). Since all resulting validity statements are derived from these validity degrees, the refinement of the competence estimation leads to improved results of the entire technology. In fact, the consequence of better validity statements is a "more dependable" system after its refinement. Furthermore, this competence estimation is useful for selecting appropriate experts for the panel of upcoming sessions.
2. Second, the VKB can support the identification of the optimal solution, which is the basis for the system refinement and the updating process of the VKB itself. If several solutions are candidates to be the "optimal solution", i.e. they receive the same approval by the expert panel, the information in the VKB is helpful to differentiate these candidates.

Both approaches are introduced in (Kurbad 2003).

Competence Estimation of the Experts

Since the competence estimation of the experts is based on the experts' performance in the rating session, the ratings for the test cases originated from the VKB need to be included in the estimation.

Since the VKB holds knowledge about the experts' competence in previous sessions, i.e. "historical competence", it opens the door to selecting an appropriate expert panel for a scheduled session. Derived from the information in the VKB, we formally introduced

1. a so–called *historical session competence* $sess\_est_{hist}(e_i, S'_i)$ of a certain expert e_i within a session S'_i,
2. a *historical competence trend* $trnd\_est_{hist}(e_i)$, which describes the development of an expert's e_i competence over time,
3. a *competence gain* $\Delta sess\_est_{hist}(e_i, \sigma_i^t)$ from one session to the next and an *average competence gain* $\delta_i(\sigma_i^t)$ over time,
4. a classification of experts as those with an *increasing*, *even*, and *decreasing* competence over time, and
5. an *average historical competence* $avg\_est_{hist}(e_i)$.

Finally, we developed guidelines to use these concepts for the selection of an appropriate expert panel.[5]

Identification of the Optimal Solution

For the final *system refinement* step of the entire validation technology, the concept of an optimal solution was introduced in (Knauf, Gonzalez, and Abel 2002). This is, loosely speaking, the solution $sol_{opt}(t_j)$ to a test case input t_j that gained the maximum approval by the experts in the current panel. Unfortunately, it may be that there are several solutions that enjoy the maximum approval. In these cases, the VKB is used to identify one of them as the "very best" one.

For this purpose, we introduced a step–by–step filtering process that is applied until one candidate solution is left over:

1. First, the average competence of the experts who are in the VKB's *list of supporters* of the candidate solutions are considered. The candidate solution, which enjoys the maximal support by the VKB, is considered the "very best" one.

[5]Note, that the authors themselves claim to utilize these estimations with care, because they are based on data, which might be incomplete, irrelevant, or not representative. Furthermore, social reasons require the handling of the concepts about an expert's competence with care, discretion and social responsibility.

2. In case there are still several solutions as the outcome of the step above, a *list of vetoers*[6] is derived from the VKB and their average competence is calculated by using the VKB. The candidate solution that received the minimal "resistance" by the VKB is considered the "very best" one.
3. If there are still several candidate solutions after these two steps, the supporters for each of the remaining candidate solutions are compared. The solution that is supported by the expert e_i with the maximal competence $cpt(e_i, t_j)$ for the test data t_j, is considered the "very best" one.
4. The last opportunity to identify the "very best" solution, in case there are still several ones after these three steps, we perform a "run–off" session with the expert panel and the remaining candidate solutions.

Gaining Experience in the VKB

Since these beneficial uses of the VKB are only as good as the information in it, the updating and maintenance process needs to be optimal. In particular, as much as possible, information has to be utilized for gaining validation knowledge. This information has to be processed to fit within the VKB's structure. Therefore, concepts used to maintain the VKB are introduced (Kurbad 2003).

Handling Incomplete Sessions

It might happen that some experts are no longer available during a current experimentation. Obviously, such incomplete sessions can influence the results of the entire technology. An easy way to address this issue is to exclude all information given by such experts, thus practically reducing the expert panel and wasting valuable expertise. We suggest to also take *incomplete experimentation results* into account, since the invested *human workload* is a costly and valuable factor.

There are four possible scenarios where an expert would not finish his/her work:

1. He/she leaves the panel during the *solving session* and does not return for the *rating session*,
2. he/she finishes the *solving session* properly and does not take part in the *rating session*,
3. he/she finishes the *solving session* properly and leaves the panel during the *rating session*, or
4. he/she leaves the panel during the *solving session* and returns for the *rating session*.

Case 1 and 2 can be treated in almost the same way. All identified solutions provided by the expert should be rated by all other experts, since they might contribute correct information. The competence estimations of an expert e_i are restricted to the consideration of the other experts' opinion about e_i-s competence and the self estimation of e_i while solving the test cases (Knauf, Gonzalez, and Abel 2002).

Generally, two failures may occur during the assessment of *local validity degrees* to solutions for a test input t_j:

- The validity degree assessed to the correct solution might be too low and/or
- wrong solutions might receive a validity degree that is too high.

Although both failures endanger the correct result of the experimentation, the first one seems to be less harmful. Since the local validity of a solution is weighted by the local competence of the (rating) experts, it seems preferable to underestimate a competence level than to overestimate it. Therefore, we assume that the "missing" components of competence estimation as introduced in (Knauf, Gonzalez, and Abel 2002) have a value of 0.

In **case 3**, the the other experts' opinion about e_i 's competence can be estimated for all test data t_j. Following the decision to prefer underestimation, the values of the other four components get a value of 0 for each test case that the expert did not rate.

For all other test cases, i.e. the ones that were rated by the expert, the original equation of (Knauf, Gonzalez, and Abel 2002) applies.

Case 4 can be treated in different ways:

1. A pragmatic way is to disallow the return of an expert to the panel, if he/she did not finish the solving session. Therefore, the second scenario would apply.
2. One could allow an expert only the rating of test cases that he/she solved in the solving session. Thus, that expert does the whole experimentation with a reduced number of test inputs. All equations apply as usual, but based on the reduced test case set. In case the expert does not finish the rating session, the assumptions for the third scenario will apply within this reduced experimentation.
3. Providing the possibility to rate all test cases obviously leads to more information. Assuming that the expert might also leave the rating session unfinished, three situations are imaginable:
   (a) Each test case that has been both solved and rated the "usual" way (Knauf, Gonzalez, and Abel 2002) is used for competence estimation.
   (b) For test cases that has been only solved but not rated the first and second scenario applies.
   (c) For each test case that has not been solved but rated, only a *certainty estimation* (Knauf, Gonzalez, and Abel 2002) can be performed and all other estimation values are assumed to be 0.

Obviously, with the first two variants, much of the invested human workload is wasted. Therefore, we prefer the third variant, which uses all human inputs and underestimates the experts' competence.

Maintenance of the VKB

To ensure that the VKB really gains experience while being used, it has to be updated within each validation session. Updating, in this context, means adding new cases to the VKB.

One might argue that deleting outdated "historical knowledge" has to be a part of the VKB maintenance as well.

[6] Vetoers are experts, who provided a negative rating for a considered solution.

After a long discussion, the authors reached the conclusion, that the deletion of cases should not only be avoided, but even prohibited.

Of course, humans do forget parts of their "historical knowledge" and this is considered a natural and healthy process. Often, it is even necessary, since humans retrieve historical knowledge by setting it in the actual context, which might lead to wrong conclusions.

The VKB, on the other hand, stores the historical cases explicitly and associated to the right (historical and topical) context by marking it with a time stamp and a domain description. Thus, it provides the opportunity to avoid misinterpretations. Since historical knowledge from the VKB is always revalidated within the current session by newly rating it, invalid facts are sorted out by utilizing the meta–knowledge[7] of the human experts.

The "experience" of a session, which is worth keeping, is the optimal solution sol_{Kj}^{opt} to each test data t_j that has been solved. Additionally, the associated list of solvers E_K and the list of raters E_I needs to be kept. Furthermore, a time stamp has to be provided for each new case of the VKB. The time stamp τ_S of the current experimentation session is assumed to be the starting time of the rating session. The only requirements time stamps have to meet is that they have to be determined in the same way in every session to maintain their order over time. By adding a description of the application domain and a context D_C, all resulting 8–tuples $[t_j, E_K, E_I, sol_{Kj}^{opt}, r_{IjK}, c_{IjK}, \tau_S, D_C]$ have to be stored as new elements of the VKB.

Validation Expert Software Agents

With the view to future opportunities for replacing human input, the VKB itself is extended by a Validation Expert Software Agent ($VESA$) concept. $VESA$s obtain and store validation knowledge / data autonomously from validation results of the experts participating in the test case experimentation.

In fact, the $VESA$ concept adopts the idea of software agents in general and the recent developments in this field (Weiss 1997). In particular, Singh and Huhns (1997) address some basic concepts and assumptions as used here as well. However, advanced ideas like

- the issue of learning,
- the issue of cooperation and competition, and
- the issue learning
  - about/from other agents and the world or
  - by communication and understanding

are far away from the fundamental agent concept introduced here.

The basic assumption of our agent concept is that experts who provide similar solutions to test cases and similar ratings to other experts' solutions might have a similar knowledge structure. Therefore, an expert might be modelled by an agent that provides the response of another human expert, who had a maximum similarity with the considered expert in the past.

Each $VESA$ is an autonomous software agent corresponding to a particular human expert. It gains personal validation knowledge mainly from personal data such as (not always best) solutions, ratings, etc. of the human expert validator corresponding to it. Furthermore, is can be considered to be a model that represents the validation experience and behavior of a group or an organization of validation experts.

In every validation session, the $VESA$s become more intelligent as well as more adaptive to wider (similar but slightly different) applications, since they can learn from test inputs, the associated answers, their certainties and their ratings provided by the human validators. Namely, they increase their validation competence through validation knowledge gained by various sessions over time.

Though a $VESA$ is a model of a human validation expert, it can also gain the validation knowledge / data of other validators, when a very high–rated (but not always best) solution happens to be derived by one of the same type of validators which usually have almost the same solutions. Since they are not human but machine, anonymity will be kept even if they get information from other (human) experts. They do not need the name of each expert, but rather an ID to distinguish whether or not the information belongs to the same expert.

Sources of VESA's Knowledge

The knowledge base to dynamically form a $VESA$ in case of its need is simple: Gaining all information that is available. For each human expert it keeps (1) each and every solution he/she provided to a test data, (2) each and every rating he/she provided to a solution in (3) each and every historic session indicated by a time stamp.

Dynamic Construction of VESA

In case an expert e_i is not available to solve a case t_j, e_i-s former (latest) solution is considered. It is assumed that e_i has still the same option about t_j-s solution. Thus, $VESA$ provides it.

If e_i never saw a case like t_j before, similarities with other experts (which might have the same "school" or "thinking structures") are considered. Among all experts, who ever delivered a solution to t_j, the one with the largest subset of the solutions like e_i-s to the other cases is identified as the one with the most similar behavior. His/her solution is assumed to be the one of e_i as well, and thus provided by $VESA$.

In case a $VESA$ is requested to provide a rating to a given solution, similar considerations lead to an "assumed rating" of e_i:

1. If e_i considered (solved or rated) the same test case t_j in former sessions, we look at the rating or the provided solution with the latest time stamp: In case the latest consideration is a rating, both the same rating r and the same certainty c are adopted and provided by $VESA$. In case the latest consideration is a provided solution sol (different from "unknown"), $VESA$ provides for this solution

[7] Meta–knowledge is "knowledge about knowledge", i.e. about its retrieval, context, usage, etc.

a rating $r = 1$ (correct) and a certainty $c = 1$ (for sure) and for all other solutions a rating $r = 0$ (incorrect) and a certainty $c = 1$.

2. If e_i never considered (solved or rated) the test case t_j in former sessions, we look for a "most similar" expert e_{sim} who solved this case, i.e. a one who provided the largest amount of the same solutions and/or ratings to other cases in the past. If the latest consideration of t_j by e_{sim} is a rating r along with a certainty c, $VESA$ adopts and provides both. If the latest consideration of t_j by e_{sim} is a solution sol, $VESA$ provides for this solution a rating $r = 1$ (correct) and a certainty $c = 1$ (for sure) and for all other solutions a rating $r = 0$ (incorrect) and a certainty $c = 1$.

As a future benefit of the $VESA$s we expect that

1. $VESA$ can replace the human expert when he/she is too busy or too expensive to participate in validation,
2. $VESA$ can be a competent validator and upgrade the test case experimentation and
3. a group of $VESA$s might do test case experimentation without experts, since they have different validation knowledge and can be tested from various views.

Therefore, the $VESA$ concept brings a really new dimension into the validation technology by displacing human input systematically to software agents.

In fact, to learn a model of the human experts' problem solving behavior, $VESA$ still depends on the knowledge of human validators. Learning in the concept of $VESA$ is analyzing the solving and rating behavior of human their origins. The quality of the learning results, i.e. the quality of $VESA$, depends on the quantity and coverage of data provided by the human experts. Therefore,

- on the one hand, a $VESA$ is able to replace its human origin temporary, but,
- on the other hand, a $VESA$ becomes worse in case of missing human input over a long period.

Summary and Outlook

The following statements summarize the basic messages of the present paper:

1. AI system validation technologies so far are time consuming, expensive, and depend on an undependable resource, the "human expert".
2. The VKB concept is the key to using this resource efficiently.
3. While VKB aims at modelling the human experts' collective and most accepted (best rated) knowledge, $VESA$ aims at modelling a particular human expertise.
4. At some point (after learning an appropriate model) $VESA$ allows the replacement of its human origin.
5. Experiments to test the VKB concept prototypically are subject of our actual research.
6. The $VESA$ concept is both still a subject of conceptual research and discussion and already a subject of first empirical experiments.
7. A new dimension is seen in validating the $VESA$ concept by comparing their solutions and ratings with the ones of its human original in case of availability.

References

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

Knauf, R.; Gonzalez, A.J.; Tsuruta, S. 2003. Utilizing Validation Experience for System Validation. I. Russell & S. Haller (eds.): *Proc. of 16th International Florida Artificial Intelligence Research Society Conference 2003 (FLAIRS-2003)*, St. Augustine, FL, USA, pp. 223-227, Menlo Park, CA: AAAI Press.

Kurbad, T. 2003. A Concept to Apply a Turing Test Technology for System Validation that Utilizes External Validation Knowledge. Diploma Thesis, Ilmenau Technical University, Faculty of Computer Science and Automation, Inv.-Nr. 2003-09-03/053/IN95/2238

Tsuruta, S.; Onoyama, T.; Kubota, S; Oyanagi, K. 2000b. Knowledge-based Approach for Validating Intelligent Systems. Kern (ed.): *Proc. of* 45^{th} *Internat. Scientific Colloquium (IWK-00)*, Ilmenau, Germany, Technical Univ. of Ilmenau, pp. 769–774.

Tsuruta, S.; Onoyama, T.; Taniguchi, Y. 2002. Knowledge–Based Validation Method for Validating Intelligent Systems. Kohlen (ed.): *Proc. of the* 15^{th} *Internat. Florida Artificial Intelligence Research Society Conference 2002 (FLAIRS-02)*, Pensacola Beach, FL, USA, pp. 226–230, Menlo Park, CA:AAAI Press.

Weiss, G. (ed.). 1997. *Distributed Artificial Intelligence Meets Machine Learning: Learning in Multi-Agent Environments*. Berlin: Springer-Verlag, Lecture Notes in Computer Science Volume 1237, ISBN 3-540-62934-3.

Singh, M.; Huhns, M. 1997. Challenges for Machine Learning in Cooperative Information Systems. in (Weiss 1997), pp. 11–24.