Knowledge-based Validation Method for Validating Intelligent Systems

Setsuo Tsuruta*, Takashi Onoyama** and Yoshio Taniguchi***

* Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji Asao-ku Kawasaki-shi, 215-0013, Japan
e-mail: tsuruta@sdl.hitachi.co.jp
** Hitachi Software Engineering Co.,Ltd
6-81,Onoe-cho,Naka-ku,Yokohama,231-8475,Japan
e-mail: onoyama@ori.hitachi-sk.co.jp
***Hitachi Microsoftware Systems, Inc.
Yoshida-cho Totsuka Yokohama, 244-0817, Japan
tanigu@hitachi-ms.co.jp

Abstract

To diminish the validation load on busy experts, a bi-directional, many-sided explanation typed multi-step validation method (MMBV) was proposed, which can decrease the validation tasks of experts (Tsuruta et al., 2000).

This paper proposes the extension of MMBV. In this method, through the validation results and comments of experts, validation knowledge can be automatically acquired as case-based knowledge. Utilizing this knowledge, computers and KEs can validate, mostly without experts.

Consequently, this method is effective for solving the bottleneck in acquiring validation knowledge from experts who are busy, and this further reduces validation load of such experts.

1. Introduction

Validation methods for ordinary software systems (Myers, 1979) have made considerable progress by focusing on the detection of programming bugs. However, as to intelligent systems' validation, though methods for validating expert systems have been proposed (Terano, 1992), they mention almost nothing more than general check items. The Turing Test approaches seem promising in validating intelligent systems (Turing, 1950), (Knauf et al., 1998), but they impose excessively heavy burdens and responsibility on experts.

Experts are very busy. Still more, they are anxious about losing their vocational superiority due to the appearance of practical intelligent systems. To decrease the validation load on such experts, and to obtain their cooperation more easily, a multi-step many-sided explanation typed bi-directional validation method (MMBV) has been proposed (Tsuruta et al., 2000). Utilizing MMBV, experts, KEs and computers can share the validation load.

In order for KEs and computers to share more validation load with experts, it is important for KEs and computers to share more validation knowledge with experts, and for validation knowledge to be incorporated in computers. However, there is a serious problem called "knowledge acquisition bottleneck". Validation knowledge is different from domain knowledge. It can be rather more difficult than domain knowledge since validation knowledge is used for validating domain knowledge.

To solve this problem, the extension of MMBV is proposed. In this new method, mainly computers supported by KEs acquire, represent, validate and refine knowledge for validation, based on experts' validation data. And, KEs can further share the validation load of busy experts, through checking and modifying the results of automatic validation by computer's inference utilizing the validation knowledge acquired and incorporated in computers. This method is considered to be effective for solving the bottleneck of acquiring validation knowledge through busy experts and for reducing experts' load in validation itself as well as validation knowledge acquisition.

Section 2 describes problems and previous proposals in intelligent systems' validation. Section 3 describes the new validation method. In section 4, this method is evaluated and Section 5 makes concluding remarks.

2. Previous Proposal for Validating Intelligent Systems and its Problems

2.1 Previously proposed Validation Method

MMBV has been proposed to decrease the validation load on experts and to obtain their cooperation more easily (Tsuruta et al., 2000), since experts are busy and anxious about losing their vocational superiority through being substituted by intelligent systems (Tsuruta et al., 1997).

This method (see fig.1) has three characteristics:

1) Multi-step validation comprising three steps, namely, automatic validation, validation by KEs (Knowledge Engineers) and validation by experts.
2) Many-sided explanation types to multilaterally edit and explain test results of a solving method in each phase in the process, including elemental methods of

the solving method, in forms required at each of the foregoing steps such as statistical data, charts, graphs, topological maps and real maps.

3) Bi-directional interaction under a distributed information environment in each of foregoing steps using many-sided explanations and other factors.

By utilizing this system, experts can concentrate on areas which cannot be decided by computers or KEs, and can make efficient validation by utilizing bi-directional multi-step and many-sided explanation types.

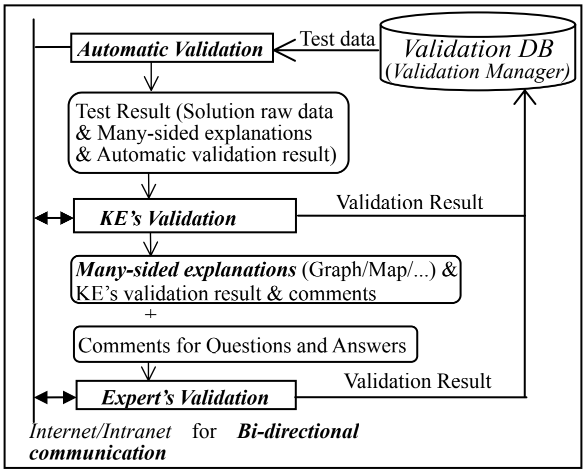
Automatic Validation
Test data
Validation DB
(Validation Manager)
Test Result (Solution raw data
& Many-sided explanations
& Automatic validation result)
KE's Validation
Validation Result
Many-sided explanations (Graph/Map/...) &
KE's validation result & comments
+
Comments for Questions and Answers
Expert's Validation
Validation Result
Internet/Intranet for Bi-directional communication

Fig.1 Bi-directional Many-sided Explanation Typed Multi-step Validation Method

2.2 Knowledge Acquisition Problems in Validation

In order for KEs and computers to share validation load with experts, it is important for KEs to share validation knowledge with experts, and for validation knowledge to be incorporated in computers. However, there is a difficulty called "knowledge acquisition bottleneck".

Validation knowledge is different from domain knowledge. It can be rather more difficult than domain knowledge since validation knowledge is used for validating domain knowledge incorporated in intelligent systems. It includes global / social / political factors which are complex and intuitive. As an example, for a large distribution network, experts intuitively validate the scheduling expert systems by checking the shape and other factors of several generated track routes.

Experts are too busy to analyze and explain such validation knowledge. Further, only a few experts readily explain such validation knowledge. Since experts to build a system are apt to believe their knowledge incorporated in it is correct, they had better not validate this system, however cooperative they are. Thus, other cooperative experts are necessary, but many experts dislike to be replaced by intelligent systems and are uncooperative.

Yet, validation knowledge or validation results are often inconsistent or different depending on experts.

Thus, it is very difficult to acquire objectively correct validation knowledge. To acquire objectively correct validation knowledge, seemingly, we cannot depend on interviews with experts very heavily.

3. Knowledge-based Method for the Validation of Intelligent Systems

3.1Concept of Knowledge-based Validation Method

To solve the foregoing problems, an advanced method of MMBV is proposed. In this new method, mainly computers checked and supported by KEs acquire, represent, validate and refine validation knowledge, based on the experts' validation data. Thus, precious validation knowledge can be acquired and incorporated as correct and consistent knowledge-base, though such knowledge is difficult to acquire because experts are very busy and is often different or inconsistent depending on experts. Computers can automatically infer the validation results, utilizing the acquired validation knowledge, and can further share the validation load of busy experts, with the help of KEs who check and modify the automatic validation results.

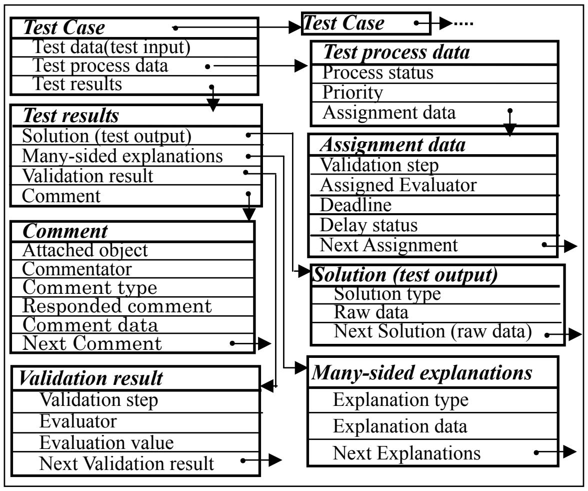

Fig. 2 Structure of data in validation DB

3.2 Features of the new Validation Method

The following is the characteristics of this new method:

1) Learn by experts' validation data

In this new method, the validation knowledge is acquired through experts' validation data including "validation result" and "comment" of experts. Experts' validation data are stored in the validation DB (VDB) of the validation system based on MMBV (fig. 1).

The data structure of VDB is shown in Fig. 2. VDB mainly consists of test cases. A test case consists of Test data, Test process data and test results. Test process data includes Assigned Evaluator, (validation) Deadline and Delay status. Test results consist of Solution (raw data), Many-sided explanations, Validation results, and Comments. Solution (raw data) includes the solution itself. Many-sided explanation consists of Explanation type to indicate the kind of explanations (e.g. Solving step, Statistical value, Graph, Table) and explanation data. Validation result includes Evaluator and Evaluation value such as {OK (valid), NG (invalid)}. Comment includes Commentator, Responded comments, and Comment data, namely, contents represented in natural language.

Thus, experts' validation data in VDB are experts' validation results and comments, accompanied with test inputs (problems or test data) and test results (solutions and many-sided explanations including explanations of the relations between test inputs and outputs).

Referring to such experts' validation data, validation knowledge is automatically constructed and stored in validation knowledge base (VKB). Many-sided explanations are used for KE's understandability in checking / validating the validation knowledge.

2) Validation knowledge representation and the acquisition of automatic validation knowledge

Experts' validation data mentioned in 1) stored in VDB are considered as experts' validation examples or cases. Thus, the validation knowledge is represented as case-based knowledge (Kolodner, 1999). Namely, by way of putting the test inputs (problems) and the test outputs (solutions) into the case-condition (case-problem) part, and experts' validation results (expert's evaluation or, estimation value) and comments into the case-solution part, the validation knowledge represented by cases can be constructed through experts' validation data.

For example, as for a TSP (Traveling Salesman Problem), a case-condition (case-problem part) is described as a list of cities and constraints (test inputs or a test problem), accompanied with test outputs (a test solution) such as an ordered sequence of cities to be visited. And a case-solution part is the expert's evaluation (estimation) value.

A case has also various properties such as the confidence value (CV), many-sided explanation, expert's comment, and SUPPORTER, the list of supporters who support the case. Further, in order to confirm the correctness of acquired case base knowledge, each case is linked with VDB 's test data and test results called "corresponding validation data" through which the case was constructed.

The idea of this knowledge acquisition procedure is as follows.

<1> From VDB, fetch the experts' validation data when expert has just validated. Construct an applicant (of a new) case as mentioned above with its CV being 1 and its SUPPORTER being the ID of the experts described in Evaluator of Validation result in VDB. And link the applicant case with experts' validation data in VDB as applicant's corresponding validation data.

<2> If the same case exists in the validation knowledge base (VKB), the CV is increased by 1, and the applicant is not added to the VKB (Integrated).

<3> If, in VKB, there exists a case whose case-condition part is the same but case-solution part is different, for confirmation, the experts' validation data linked with this existing case is immediately presented to the expert indicated in the applicant's corresponding validation data. If he changes the evaluation value of the validation result in the applicant's corresponding validation data, the applicant is not added to the VKB, but the CV of the above case existing in VKB is increased by 1. If he does not change the above evaluation value corresponding to the applicant, inconsistency occurs. In such cases, the applicant is not added to the VKB, and later, the retry procedure described in 4) is started.

<4> If there exists a case whose case-condition part is the same as the applicant only in the problem or test inputs and whose case-solution part is "valid" and the same as the applicant, because of doubt, the experts' validation data of the existing case is immediately presented to the expert. This is done for confirmation. Here, the first condition means that the solution or test output is different from the applicant. And the expert's name or ID is indicated in applicant's corresponding validation data. If he changes the evaluation value of the validation result concerning the applicant, the applicant is not added to the VKB. Otherwise, the applicant case is added to the VKB. If he doubts the evaluation value of the validation result in the validation data corresponding to the existing case in VKB, the retry procedure is started later as in 4). The new applicant is confirmed already and not retried.

<5> Otherwise, the applicant is added to the VKB, as a new case with the CV being 1.

The automatic validation can be done through computer's inference, utilizing validation knowledge acquired and represented in the above way.

3) Acquisition of interactive validation knowledge

Furthermore, each of experts' comments can often be considered as a case. Such comments can be used in order to construct or specialize/generalize existing cases. This kind of validation knowledge (cases) can be acquired through KEs' analysis of experts' comments (e.g. since this road is passing through a park where citizens enjoy playing sports, music etc., the distribution network should not include this road section). This case-based validation knowledge is used by KEs under the support of computers in searching related cases or in automatically checking if the solutions (test results) are different from those described in case-solution part of the validation knowledge represented by cases. Through analyzing experts' comments, KEs acquire and incorporate knowledge such as one for excluding prohibited road sections from solutions.

4) Validation of validation knowledge and its refinement and management

For correct validation of the intelligent system, it is necessary to validate the above-acquired validation knowledge. In the newly proposed method, the automatically acquired new validation cases (applicants, strictly speaking) are checked immediately by experts (linked with the applicant cases) in comparison with the existing validation cases, as explained in 2). If inconsistency or doubt occurs as described in <3>, <4> of 2), a validation manager (VM) or a KE starts the following retry procedure, not immediately but after allotting experts to retry the validation, considering their load. Here, the expert allotted for retry is the third expert who is different from either the expert concerning the applicant case or another concerning the existing case.

<1> Retry for inconsistency in <3> of 2):

If the third expert supports only the applicant, the existing case is deleted but linked with the applicant that is added to VKB.

If the third expert supports only the existing case, the applicant is not added to the VKB. The CV of the existing case is increased by 1, and the third expert's ID is added to its SUPPORTER for later reference.

If the third expert supports both the applicant case and the existing case, the VM or KE asks all experts concerning both cases to comment on the conditions for discriminating among these cases, and modifies both applicant and existing cases according to the comment. The modified applicant is added to the VKB. The CV of the modified existing case is increased by 1, and the third expert's ID is added to its SUPPORTER for later reference.

If the third expert does not support both the applicant case and the corresponding existing case, the applicant is not added to the VKB. The CV of the existing case is decreased by 1

<2> Retry for doubt in <4> of 2):

As mentioned in 2), only the existing case is retried.

If the third expert supports the existing case, the CV of the existing case in VKB is increased by 1, and the ID of the third expert is added to its SUPPORTER. Here, if there are modification requests from some of the experts concerning the case, the VM or KE asks all related experts to comment about the modification, and modifies the existing case according to their comments.

If the third expert does not support the existing case, the applicant is not added to the VKB. The CV of the existing case is decreased by 1.

Meanwhile, as to interactive validation knowledge acquired by KE, computers and KEs interactively check the experts' validation data. If inconsistencies are found, retry is done also here. If inconsistencies still occur, KEs ask experts, using bi-directional communication.

This method is also effective for refining validation knowledge through adding necessary conditions to each of this knowledge. The reason is because experts' validation results are often different on various situations, which are not easily clarified as symbolic conditions and not exactly the same as those of other similar validation knowledge. However, due to this method, such situations can possibly be clarified as case-condition parts of case-based validation knowledge through computer's automatic comparison (search and check) and KE's analysis, though experts may have to help when needed.

As to the management of validation knowledge, each piece of knowledge has the order according to its CV and age (oldness indicated by time stamped on cases). The case whose CV is zero or below zero is removed from VKB immediately. The oldest case, whose case-solution part's value is "invalid" and whose CV is the lowest, is removed from VKB, when no space exists for VKB.

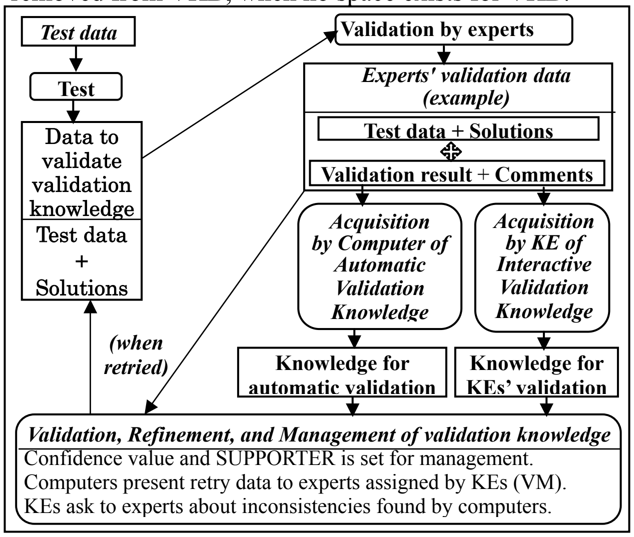
Test data
Test
Data to validate validation knowledge
Test data + Solutions
Validation by experts
Experts' validation data (example)
Test data + Solutions
Validation result + Comments
Acquisition by Computer of Automatic Validation Knowledge
Acquisition by KE of Interactive Validation Knowledge
(when retried)
Knowledge for automatic validation
Knowledge for KEs' validation
Validation, Refinement, and Management of validation knowledge
Confidence value and SUPPORTER is set for management.
Computers present retry data to experts assigned by KEs (VM).
KEs ask to experts about inconsistencies found by computers.

Fig.3 Acquisition, Validation, Refinement and Management of Validation Knowledge

Thus, validation knowledge is acquired, validated, refined, and managed as shown in Fig. 3.

4. Evaluation

4.1 Other Validation Methods and Tools

Many excellent theories and ideas based on well-known Turing Test (Turing, 1950) have been proposed as methods to validate AI systems in order to determine whether or not problem-solving knowledge (solving methods) is at expert level. All these methods validate in a black-box style merely examining inputs and outputs (Knauf et al., 1998), (Abel & Gonzalez, 1997), (Knauf & Gonzalez, 1997). Compared with our proposed method, the validation load on experts is not considered in these methods.

Solution optimality and response time of intelligent solving methods, which use heuristics and knowledge, are greatly different depending on the problem pattern (Tsuruta et al., 2000). In terms of product liability, validation of their stability is mandatory. However, the number of test cases tends to increase significantly in the above mentioned black-box style or Turing type tests.

The method of reducing the number of test cases by using formally described domain models and evaluation criteria is interesting (Abel & Gonzalez, 1997). One problem with this method, however, is that man-hours are needed to create a formal model and evaluation criteria so that it cannot be easily used in practical applications. Practically, test cases are narrowed down, using informal specifications and evaluation criteria. Nevertheless, even when using the validation method, as product liability is involved, these intelligent solving methods must be validated with a significantly large number of test patterns. Therefore, loads on experts are critical.

MMBV has been proposed to decrease the validation load on busy and usually uncooperative experts. MMBV enables the validation tasks to be shared among experts, KEs and computers (Tsuruta et al., 2000). However, there is knowledge acquisition bottleneck to acquire and incorporate validation knowledge. Thus, MMBV could not exceed a limit of load sharing among computers, KEs and experts in validation tasks.

4.2 Characteristics beyond others

Our new method is considered to be effective for solving the bottleneck of validation knowledge acquisition. As to the bottleneck for acquiring problem solving knowledge, an interesting approach through the knowledge refinement using validation data (Knauf & Philippow, 2000) is proposed. Problem solving knowledge is represented as rules in their approach. However, even their approach does not say how to solve the bottleneck of validation knowledge acquisition, and does not have our method's effect for computers or KEs to share much validation load with experts. Thus, our method is expected to diminish the validation load of busy experts as well as experts' load on interviews required by KEs to acquire validation knowledge.

5. Conclusions

An advanced method of MMBV was proposed. In this new method, computers and KEs acquire, represent, validate and refine validation knowledge, based on the validation results and comments of experts. Especially, in this new method, computers can automatically acquire and manage such validation knowledge represented as cases.

Owing to this method, experts' validation knowledge can be acquired and incorporated as correct and consistent case-base, though such knowledge is difficult to acquire because experts are very busy, and is often inconsistent or different depending on experts. Computers can automatically validate utilizing the above acquired and incorporated validation knowledge, and KEs can check / modify the automatic validation results in cooperation with experts if necessary. Thus, computers and KEs can further share the validation load of busy experts and can make validation more reliable. Furthermore, KEs can use the above acquired validation knowledge for their validation.

Consequently, this method is considered to be effective for solving the bottleneck in acquiring validation knowledge from experts who are busy, and for reducing validation load of such experts.

References

Myers, G.J. 1979. The Art of Software Testing. John Wiley & Sons.Inc.

Terano, T. 1992. Expert System Evaluation Manual, Tokyo:Ohmsha.

Turing. A. M. 1950. Computing machinery and intelligence, Mind, 59:433-460.

Knauf, R.; Jantke, K. P.; Abel, T., and Philippow, I. 1998. Fundamentals of a TURING Test Approach to Validation of AI Systems. In Proceedings of 42nd International Scientific Colloquium, Technical Univ. of Ilumenau.

Tsuruta, S.; Onoyama, T.; Kubota, S; and Oyanagi, K. 2000. Validation Method for Intelligent Systems: In Proceeding of the 13th International Florida AI Research Society Conference, Orland, FL.

Abel, T.; and Gonzalez, A.J. 1997. Enlarging a Second Bottleneck: A Criteria-based Approach to Manage Expert System Validation Based on Test Cases. In Proceedings of the 10th International Florida AI Research Society Conference, Daytona Beach, FL.

Knauf, R.; and Gonzalez, A.J. 1997. Estimating an AI system's validity by a TURING Test. In Proceedings of 42nd International Scientific Colloquium. Technical Univ. of Ilumenau.

Kolodner, J. 1999. Case-Based Reasoning. Morgan Kaufmann Publishers, Inc.

Tsuruta, S.; Eguchi, T.; Yanai, S.; and Ooshima, T. 1997. Dynamic Goal centered Coordination AI system: Application and the validation problem, In Proceeding of 42nd International Scientific Colloquium, Vol.2, 52-58.Technical Univ. of Ilumenau.

Knauf, R., and Philippow, I. 2000. Towards Validation of Rule-Based Systems – The Loop is Closed. In Proceedings of 13th International Florida AI Research Society Conference, Orland, FL.