Knauf, Rainer; Tsuruta, Setsuo:

A case–based approach to explore validation experience

Publikation entstand im Rahmen der Veranstaltung:
19th International Florida Artificial Intelligence Research Society Conference 2006 (FLAIRS-2006)
11. - 13. Mai 2006, USA, Florida, Melbourne Beach

A Case–Based Approach to Explore Validation Experience

Rainer Knauf
Faculty of Computer Science
and Automation
Ilmenau Technical University
PO Box 100565, 98684 Ilmenau, Germany

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

Abstract

The success of TURING Test technologies for system validation depends on the quality of the human expertise behind the system. As an additional source of human experts' validation knowledge a Validation Knowledge base (*VKB*) and so called Validation Expert Software Agents (*VESA*s) revealed to be useful. Both concepts aim at using collective (*VKB*) and individual (*VESA*) experience gained in former validation sessions. However, a drawback of these concepts were their disability to provide a reply to cases, which have never been considered before. The paper proposes a case–based data mining approach to cluster the entries of *VKB* and *VESA* and derive a reply to unknown cases by considering a number of most similar known cases and coming to a "weighted majority" decision. The approach has been derived from the k Nearest–Neighbor approach.

Introduction

Typically, the validation of intelligent systems is performed by comparing their functionality with original human validation knowledge. In contrast to verification, which aims at ensuring compliance with specifications and the absence of specific errors without executing the system, validation typically involves rigorous and extensive testing of the system by a TURING Test technology. However, experts may not always agree among themselves. The size of the test case set, the frequency of the validation exercises and the number of experts required for each such exercise can combine to pose great burdens of time and effort on human experts. Experts are a scarce resource, have limited time, and are expensive to employ. These limitations have the potential to seriously degrade a validation exercise.

To make TURING TEST validation results less dependent on the experts' opinions and to decrease the workload of the experts, a *Validation Knowledge Base* (*VKB*) was developed as a model of collective human expertise of former expert panels and *Validation Expert Software Agent*s (*VESA*) were developed as a model of individual human expertise (Tsuruta et.al. 2002; Knauf et al. 2004a). These concepts have been implemented in a validation framework (Knauf et al. 2002). To estimate the usefulness of these concepts and to reveal their weaknesses, a prototype test was performed (Knauf et al. 2005).

This test revealed a basic disadvantage of these concepts. Since they both hack back to authentic human knowledge of former validation sessions, they were not capable to provide solutions or ratings to cases that have never been considered in the past.

Although in "toy applications" with a manageable amount of test cases (like (Knauf et al. 2005)) these concepts don't suffer from this feature, it is certainly an issue in real world application fields. Even with a background of a large validation experience it rarely happens that for an actual case exactly the same one has been processed before.

According to the idea of Case–Based Reasoning, the so–called *Locally–weighted Regression* and, as far as investigated, the way human experience works, we propose a derived version of the so–called k *Nearest–Neighbor* (k–NN) data mining method (Jantke et al. 2003; Singh 1998) to bring about a decision among the k most similar cases in the case base.

In opposition to classical case–based approaches, the suggested version of the k–NN method is applied within a normalized numerical input space of cases, in which the Euclidean distance is the basis for the defined the similarity measure.

The paper is organized as follows: The next section provides a short summary about the concepts developed so far: the validation framework, *VKB* and *VESA*. Section three is a short introduction to the k*–NN* method and section four introduces its adaption towards a usability for the intended purpose. In section five we discuss requirements to a test scenario for the suggested approach and section six summarizes the results.

The Concepts of VKB and VESA so far

For a systematic validation of intelligent systems (Knauf et al. 2002) introduces a 5–step validation framework, which consists of the steps (1) test case generation, (2) test case experimentation, (3) evaluation of the experimental results, (4) validity estimation, and (5) system refinement based on the revealed invalidities.

Due to the heavy involvement of humans, the most expensive step of our 5–step validation framework (Knauf et al. 2002) is the test case experimentation step. The recently

t_j	E_K	E_I	sol^{opt}_{Kj}	r_{ijk}	c_{ijk}	τ	D_C
t_1	e_1, e_3	$[e_1, e_2, e_3]$	o_6	$[1, 0, 1]$	$[0, 1, 1]$	1	
t_1	e_2	$[e_1, e_2, e_3]$	o_{17}	$[0, 1, 0]$	$[1, 1, 1]$	4	
t_2	e_1, e_3	$[e_1, e_2, e_3]$	o_7	$[0, 0, 1]$	$[0, 0, 1]$	1	
...	...	...	...	...	...	...	...

Table 1: An example for *VKB*'s entries

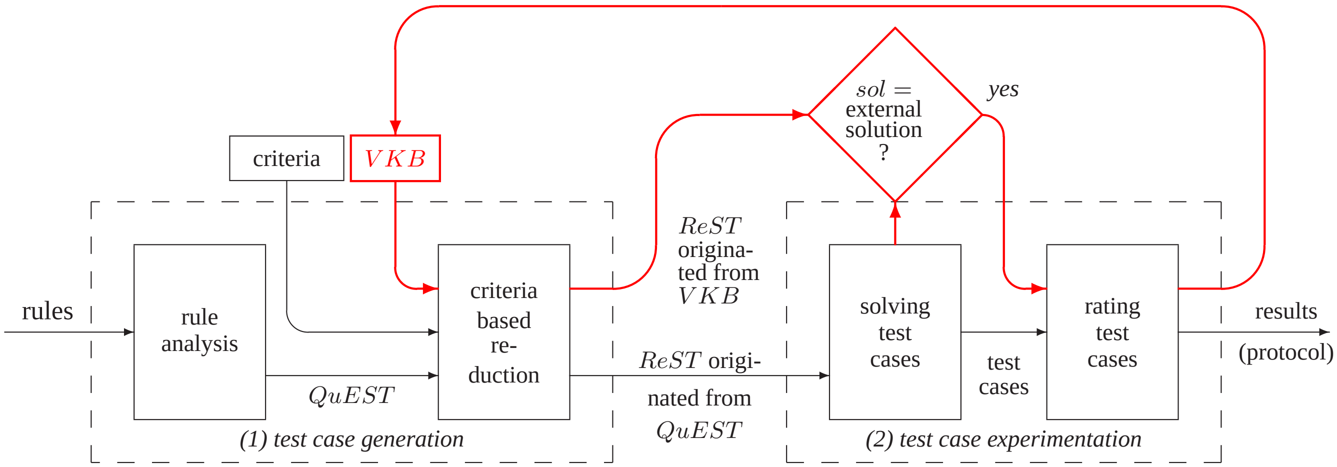

Figure 1: The use of the VKB in the Test Case Generation and Experimentation

proposed concepts of *VKB* and *VESA* aim at reducing this cost factor significantly (Knauf et al. 2005). *VKB* contains validation knowledge of previous validation processes and *VESA*s systematically model human validators by keeping the personal validation knowledge of their corresponding experts and analyzing similarities with other experts (Knauf et al. 2005).

According to the formal settings in (Knauf et al. 2002) and (Kurbad 2003), the *VKB* contains a set of previous (historical) test cases, which can be described by 8-tuples

$$[t_j, E_K, E_I, sol^{opt}_{Kj}, r_{IjK}, c_{IjK}, \tau, D_C]$$

where

- t_j is a test case input (a test data),
- E_K is a list of experts who provided this particular solution,
- E_I is a list of experts who rated this solution,
- sol^{opt}_{Kj} is a solution associated to t_j, which gained the maximum experts' approval in a validation session,
- r_{IjK} is the rating of this solution, which is provided by the experts in E_I,
- c_{IjK} is the certainty of this rating,
- τ is a time stamp associated with the validation session in which the rating was provided, and
- D_C is an informal description of the application domain C that is helpful to explain similarities between different domains or fields of knowledge.

An example, which is a part of *VKB* in the prototype test, is shown in table 1. Here, e_1, e_2, and e_3 are particular (real) human experts, $o_1, .., o_{25}$ are test case outputs (solutions), and the time stamps are represented by natural numbers $1, .., 4$.

Figure 1 sketches how the VKB is employed in the test case experimentation, which consists of

- one session to solve the test cases by both experts and the system under validation, and
- a consecutive session to rate all upcoming solutions in an anonymous TURING test.

The purpose of a *VESA* is to model a particular human expertise in the validation process. In the validation framework proposed in (Knauf et al. 2002), human expertise is requested for two tasks:

- solving test cases in a test case solving session and
- rating (other experts') solutions to these test cases in a test case rating session.

In the test case solving session, a *VESA* is requested, if an expert e_i is not available to solve or rate a case t_j. e_i's former (latest) solution is considered by this expert's *VESA*.

If e_i never considered case t_j before, similarities with other experts who might have the same "school" or "thinking structures" are considered. Among all experts who ever provided a solution to t_j , the one with the largest subset of the solutions like e_i's for the other cases that both solved is identified as the one with the most similar behavior. e_i's solution is assumed to be the same as this other expert's. This solution is consequently adopted by the *VESA* that corresponds to the missing expert. For a formal description of a *VESA*'s solving and rating behavior, see (Knauf et al. 2005).

Formally, a *VESA*$_i$ acts as follows when requested to provide an assumed solution of expert e_i for a test case input t_j:

1. In case e_i solved t_j in a former session, his/her solution with the latest time stamp will be provided by *VESA*$_i$.
2. Otherwise,
   (a) All validators e', who ever delivered a solution to t_j form a set $Solver_i^0$, which is an initial dynamic agent for e_i: $Solver_i^0 := \{e' : [t_j, E_K, \ldots] \in \textit{VKB}, e' \in E_K\}$
   (b) Select the most similar expert e_{sim} with the largest set of cases that have been solved by both e_i and e_{sim} with the same solution and in the same session. e_{sim} forms a refined dynamic agent $Solver_i^1$ for e_i: $Solver_i^1 := e_{sim}$ with $e_{sim} \in Solver_i^0$ and $|\{[t_j, E_K, \_, sol_{Kj}^{opt}, \_, \_, \tau_S, \_] : e_i \in E_K, e_{sim} \in E_K\}| \rightarrow max!$
   (c) Provide the latest solution of the expert e_{sim} to the present test case input t_j, i.e. the solution with the latest time stamp τ_S by *VESA*$_i$.
3. If there is no such most similar expert, provide $sol := unknown$ by *VESA*$_i$.

Table 2 shows an example of a *VESA*'s solutions in a prototype experiment. The experiment was intended to compare a *VESA*'s behavior (*VESA*$_2$, in the example) with the behavior of its human counterpart (e_2, in the example) to validate the *VESA* approach. t_i are test case inputs and o_i are the outputs provided by the *VESA* respectively the associated human expert.

EK_3	solution of		EK_3	solution of	
	VESA$_2$	e_2		*VESA*$_2$	e_2
t_{29}	o_8	o_8	t_{36}	o_9	o_9
t_{30}	o_9	o_9	t_{37}	o_9	o_9
t_{31}	o_2	o_2	t_{38}	o_9	o_9
...	...	...	...	...	...

Table 2: An example for a *VESA*'s solving behavior

In the test case rating session, a *VESA*$_i$ is requested to provide an assumed rating of expert e_i to a solution of a test case input t_j, it models the rating behavior of e_i as follows:

1. If e_i rated t_j before, look at the rating with the latest time stamp τ_S, *VESA*$_i$ provides the same rating r and the same certainty c on behalf of e_i.
2. Otherwise,
   (a) All validators e', who ever delivered a rating to t_j form a set $Rater_i^0$, which is an initial dynamic agent for e_i: $Rater_i^0 := \{e' : [t_j, \_, E_I, \ldots] \in \textit{VKB}, e' \in E_I\}$
   (b) Select the most similar expert e_{sim} with the largest set of cases that have been rated by both e_i and e_{sim} with the same rating r and in the same session. e_{sim} forms a refined dynamic agent $Rater_i^1$ for e_i : $Rater_i^1 := e_{sim}$ with $e_{sim} \in Rater_i^0$ and $|\{[t_j, \_, E_I, sol_{Kj}^{opt}, r_{IjK}, \_, \tau_S, \_] : e_i \in E_I, e_{sim} \in E_I, \}| \rightarrow max!$
   (c) Provide the latest rating r along with its certainty c to t_j of e_{sim} by *VESA*$_i$.
3. If there is no most similar expert e_{sim}, provide $r := norating$ along with a certainty $c := 0$ by *VESA*$_i$.

Table 3 is an example that shows a *VESA*'s behavior in a rating session that took place within the prototype experiment. Possible ratings are 1 ("correct solution to this test case input") and 0 ("incorrect solution to this test case input").

EK_3	solution	rating of	
		VESA$_2$	e_2
t_1	o_4	0	0
t_1	o_{18}	1	1
t_2	o_{20}	0	1
...	...	...	...

Table 3: An example for a *VESA*'s rating behavior

Both concepts *VKB* and *VESA* as developed so far, rely upon the availability of an entry $[t_j, \_, \_, \_, \_, \_, \_, \_] \in \textit{VKB}$, when they are asked for a solution or rating to a test data t_j. If nobody considered t_j in any previous validation exercise, both concepts fail. This fact turned out to be a limitation on the practical value of the concepts so far. Therefore, we refined these concepts by considering available entries that are similar to t_j in case there is no entry for t_j itself.

The k–NN Method

This method presupposes, that an *object* is described by a set of n attributes that have real numbers as their values. An object has a membership to exactly one out of m classes in $V = v_1, \ldots, v_m$. So the function to be learnt by the method is $f : I\!R^n \rightarrow V$[1]. Objects along with a known function value form a set of *examples*.

A *distance* $d(x^1, x^2)$ between two objects $x^1 = [x_1^1, x_2^1, \ldots, x_n^1]$ and $x^2 = [x_1^2, x_2^2, \ldots, x_n^2]$ is defined as the Euclidian distance between these objects in an n–dimensional input space:

$$d(x^1, x^2) = \sqrt{\sum_{p=1}^{n} (x_p^1 - x_p^2)^2}$$

By having a fixed number k, the method works in its simple setting as follows. It searches the k most similar objects among the examples to a given object with an unknown class membership. The class to be learnt is the one of the majority of these k cases:

$$v = \max_{v \in V} \sum_{p=1}^{k} \delta(v, f(x_p))$$

with

$$\delta(a, b) = \begin{cases} 1 & , \quad \text{if } a = b \\ 0 & , \quad \text{otherwise} \end{cases}$$

[1]Because of irrelevance for our application, we refrain from considering the method for real-valued functions.

Figure 2 shows a two–dimensional example with $V = \{\oplus, \otimes\}$. Here, different values of k result in different classes for an object $\diamond$:

$$v = \begin{cases} \oplus & , \text{ if } k = 1 \\ \otimes & , \text{ if } k = 5 \end{cases}$$

In fact, a k that is too small bags the risk that the method

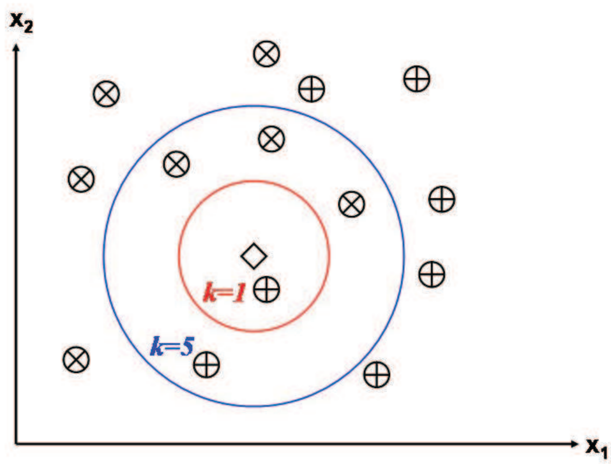

Figure 2: The influence of the parameter k

becomes too sensible to outliers. On the other hand, a k that is too large, includes too many examples from other clusters (classes). The topical literature suggests $1 << k < 10$, for example $k = 7$.

In an advanced setting, the k nearest examples $x_1, \ldots, x_k$ are weighted by their reciprocal quadratic distance to the object y to be classified:

$$v = \begin{cases} f(x_i) & , \text{ if } y = x_i \\ max_{v \in V} \sum_{p=1}^{k} \omega_p * \delta(v, f(x_p)) & , \text{ otherwise} \end{cases}$$

with

$$\delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases}$$

$$\omega_p = \frac{1}{d(y, x_p)^2}$$

If, for example, one of the k "nearest neighbors" has twice the distance of another one, its impact on the class membership a quarter of the other one.

Adapting the Method

Here we propose to define the reciprocal quadratic distance of two cases in a normalized numerical input space of test case inputs as their similarity.

In our setting, a *case* is a pair $[t_j, sol_j]$ of a *test data* t_j and its *solution* sol_j. Here the data t_j is the object and sol_j the function value we look for. The data is a vector of *test data components* $t_j = [s_j^1, s_j^2, \ldots, s_j^p]$. The example set is formed by the respective components $[t_j, sol_{Kj}^{opt}]$ of the cases in the *VKB* along with their time stamp τ.

Test data components don't have to be real–valued. Instead, they can be of different data types:

- boolean,
- a set of values with an application–driven ordering relation in–between, and
- a set of values with no (reasonable) ordering relation in–between.

Additionally, there is a time stamp τ which should be included in the similarity measure. The reason to include the time stamp in the similarity measure is modeling the learning curve. Most recent knowledge should be favored over potentially outdated knowledge.

The function values, on the other hand, are of the requested kind: exactly one solution sol_j out of m solutions $sol^1, sol^2, \ldots, sol^m$.

We feel, any similarity approach for our data in *VKB* has to meet the following requests:

1. Each test data component should influence the distance metrics to the same degree, i.e. the components have to be normalized.
2. Non–numerical test data components with in inherent ordering relation have to be enumerated to define a distance in–between them.
3. Non–numerical test data without an inherent ordering relation contribute a distance of zero in case of identity and of a maximum with respect to the normalization in case of non–identity.
4. The time stamp has to be considered a test data component as well, i.e. its $(p + 1)$ –th component to involve the time distance when computing a similarity.

Thus, we pre–process each test data component t_j as well the data of the case to be classified $t_{j\diamond}$ in a way, that it is real valued in the range $[0, 1]$. A *pre–processed test data* used for computing the distance metrics is $\hat{t}_j = [\hat{s}_j^1, \hat{s}_j^2, \ldots, \hat{s}_j^p, \hat{\tau}]$. Its components $\hat{s}_j^i$ respectively $\hat{\tau}$ are computed as follows:

- For numerical components s_j^i there is a minimum and maximum value $s_{j\ min}^i$ and $s_{j\ max}^i$ for the respective component in the *VKB*. The pre–processed component is

$$\hat{s}_j^i = \frac{s_j^i - s_{j\ min}^i}{s_{j\ max}^i - s_{j\ min}^i}$$

- For non–numerical components with an inherent ordering relation as well as for the time stamp τ all particular values in the *VKB* are consecutively enumerated by natural numbers with respect to their order, starting with 0 for the smallest value and ranging up to max for their largest value. Let n_j be the respective number of a value s_j after enumeration. The pre–processed component is

$$\hat{s}_j^i = \frac{n_j}{max} \text{ respectively } \hat{\tau} = \frac{\tau}{max}$$

- The pre–processed component for a non–numerical component s_j^i without an inherent ordering relation is

$$\hat{s}_j^i = \begin{cases} 0 & , \text{ if } s_j^i = s_{j\diamond}^i \\ 1 & , \text{ otherwise} \end{cases}$$

We adapt a commonly accepted suggestion of the data mining community[2] to choose the value of $k = 7$. We feel that with this prime value of k the risk of receiving more than one most accepted solution(s) is almost zero.

So we propose to look for the 7 most similar preprocessed test data in the *VKB* when asked for a solution or a rating to a new case. If there is no unique majority among these 7 cases, we suggest to provide the solution or rating, which has the most recent average value of the time stamp among the candidates with the same (maximum) of cases with this solution or rating.

Refining the VKB concept

If the *VKB* is asked for a solution $sol(t_j)$ to a test data t_j, it provides the most recent solution, if there is an entry for t_j in the *VKB*. If there is no such entry, *VKB* provides the reciprocal quadratic distance weighted majority solution of the 7 most similar cases:

$$sol(t_j) = \begin{cases} sol_{K_j}^{opt} & \text{, if } Entry \\ \underset{\hat{t_i} \in T}{max} \quad \sum_{i=1}^{7} \omega_p * \delta(sol_j, sol_i) & \text{, otherwise} \end{cases}$$

with

$$\begin{aligned} Entry &\equiv ([t_j, \_, \_, sol_{K_j}^{opt}, \_, \_, \tau, \_] \in VKB) \quad \wedge \\ & \quad (\neg\exists[t_j^*, \_, \_, sol_{K_j}^{opt*}, \_, \_, \tau^*, \_] \in VKB : \\ & \quad \tau^* > \tau) \\ T &= \{\{\hat{t_1}, \ldots, \hat{t_7}\} : \\ & \quad \neg\exists\hat{t^*} : d(\hat{t_j}, \hat{t^*}) < \max_{i=1,\ldots,7}(d(\hat{t_j}, \hat{t_i}) =)\} \\ d(\hat{t_j}, \hat{t_i}) &= \sqrt{\sum_{k=1}^{p}(\hat{s}_j^k - \hat{s}_i^k)^2 + (\hat{\tau_j} - \hat{\tau_i})^2} \\ \omega_p &= \frac{1}{d(\hat{t_j}, \hat{t_i})^2} \\ \delta(a, b) &= \begin{cases} 1, \text{ if } a = b \\ 0, \text{ otherwise} \end{cases} \end{aligned}$$

As a consequence of this refinement, a non–empty *VKB* will always be able to provide a solution to a given test data t_j, even if there is no respective entry in it.

However, there are still two minor disadvantages of this approach:

1. The solution provided by the *VKB* doesn't have to be an external one, because the same solution to t_j might have been provided by the system or by a human expert involved in the current validation exercise. So the *VKB* doesn't fulfill its intended purpose, which is contributing external knowledge from outside the current session.
2. It the allocation of cases in the *VKB* is not appropriate, the k nearest neighbors might be still to far to derive a solution or a rating to a considered case which is not in the *VKB*. Here, the definition of a minimum similarity might be helpful.

[2] In the Data Mining Tutor at *http//neumann.dfki.uni-sb.de/damit/* Germany's major experts in data mining have been requested to contribute the content of this e–learning system. For the k–NN method, they suggest $k = 7$.

Refining the VESA concept

So far, a *VESA* models human expertise by adapting former expertise of its original (human) expert or, if this is not available, another human expert, who solved respectively rated the considered case and behaved similar to the modeled expert in the past when handling other cases.

If there is no human expertise at all for a considered case, the *VESA* can't provide a requested solution or rating to a test case so far.

If this situation, now the refined *VESA* considers the 7 most similar cases among the solved respectively rates ones.

If the *VESA*$_i$ (the model of the human expert's e_i validation knowledge) is asked for a solution $sol(t_j)$ or a rating r along with a certainty c to a test data t_j and there is no solution respectively rating and certainty from a former exercise available, *VESA*'s reply on this request is based on the set T of the seven cases, which are most similar to t_j:

$$T = \{\{\hat{t_1}, \ldots, \hat{t_7}\} : \neg\exists\hat{t^*} : d(\hat{t_j}, \hat{t^*}) < \max_{i=1,\ldots,7}(d(\hat{t_j}, \hat{t_i}))\}$$

For deriving a solution $sol(t_j)$, *VESA*$_i$ acts as follows:

1. All validation experts e', who ever delivered a solution to any case in T form a set $Solver_i^0$, which is an initial dynamic agent for e_i: $Solver_i^0 := \{e' : [t_k, E_K, \ldots] \in VKB, t_k \in T, e' \in E_K\}$.
2. Select the most similar expert e_{sim} with the largest set of cases that has been solved by both e_i and e_{sim} with the same solution and in the same session. e_{sim} forms a refined dynamic agent $Solver_i^1$ for e_i: $Solver_i^1 := e_{sim} : e_{sim} \in Solver_i^0, |\{[\_, E_K, \ldots] : e_i \in E_K, e_{sim} \in E_K\}| \longrightarrow max!$.
3. Determine the set $VKB(e_{sim}, T) \in VKB$ of solutions to any case $t \in T$, which are supported by e_{sim}: $VKB(e_{sim}, T) = \{[t, E_K, \ldots] : t \in T, e_{sim} \in E_K\}$.
4. *VESA*$_i$ provides the reciprocal quadratic distance weighted majority solution like *VKB* does, but based only on the subset $VKB(e_{sim}, T) \subseteq VKB$.
5. If the *VKB* is too small to determine T, *VESA*$_i$ provides $sol := unknown$.

For deriving a rating r along with a certainty c, *VESA*$_i$ acts as follows:

1. All validation experts e', who ever delivered a rating to any case in T form a set $Rater_i^0$, which is an initial dynamic agent for e_i: $Rater_i^0 := \{e' : [t_k, \_, E_I, \ldots] \in VKB, t_k \in T, e' \in E_I\}$.
2. Select the most similar expert e_{sim} with the largest set of cases that has been rated by both e_i and e_{sim} with the same rating and in the same session. e_{sim} forms a refined dynamic agent $Rater_i^1$ for e_i: $Rater_i^1 := e_{sim} : e_{sim} \in Rater_i^0, |\{[\_, \_, E_I, \ldots] : e_i \in E_I, e_{sim} \in E_I\}| \longrightarrow max!$.

3. Determine the set $VKB(e_{sim}, T) \in VKB$ of ratings to any case $t \in T$, which are provided by e_{sim}: $VKB(e_{sim}, T) = \{[t, \_, E_I, \dots] : t \in T, e_{sim} \in E_I\}$.
4. *VESA*$_i$ provides the reciprocal quadratic distance weighted majority rating like *VKB* does, but
   - based only on the subset $VKB(e_{sim}, T) \subseteq VKB$,
   - by including the solutions as a $(p+2)$–th component (besides the p inputs $s^1, \dots, s^p$ and the time stamp τ), and
   - by considering the rating $r \in \{0, 1\}$ as the classes to derive by the k–NN method.

   There is a certainty c_{IjK} attached to each rating r_{IjK}. The certainty c is set to the majority of certainties (0 or 1) of the cases that derived the rating, in stalemate situations c is set to 0: $c := 0$.
5. If the *VKB* is too small to determine T, *VESA*$_i$ provides $r := norating$ along with a certainty $c := 0$.

Summary

To compensate the weaknesses and/or the unavailability of human experts for system validation, a models of both collective experience (a Validation Knowledge Base *VKB*) and individual human experience (Validation Expert Software Agents *VESA*s) have been introduced.

However, these models suffer from not providing a requested reply to cases that have never been considered by human expert panels in the past.

To overcome this drawback, the paper suggests a clustering of the available cases, which is known as a data mining method, the k nearest neighbor (k–NN) method. By this method, the entries of *VKB* and the *VESA*s are clustered and a requested reply is derived by considering a number of k most similar example cases with a known class membership.For providing solutions, the solution to test cases are considered as classes to be derived, for providing ratings, the ratings are the target of classification.

When used with an appropriate k, this method is robust against single examples with a wrong class membership. Since the *VKB* is constructed by human input, this feature is desirable.

However, some assumptions of the k–NN method are not met in our settings. Therefore, the paper introduces a method to pre–process the examples cases in the *VKB* for using the k–NN method.

Due to the nature of all data mining technologies, the quality of the *VKB* and *VESA* responses dirived by using the k–NN method heavily depends on the quantity and quality of the collected data. In fact, these data needs to have some minimum density in the input space to ensure that there are enough test cases in the same cluster to form a majority within the k nearest ones, that are weighted by their quadratic distance to a considered point. Indeed, the relationship between nature of the input space, the allocation of the data base entries, and the expected quality of the results of this method needs some more research.

Our upcoming research on this approach faces three issues:

1. an empirical evaluation of the approach by a prototype experiment,
2. the derivation of requirements to the size of the data base and the allocation of its entries to ensure that this method leads to satisfactory results,
3. a derivation of an appropriate k for successfully applying the k–NN method, and
4. a method to estimate the quality of a set of examples with respect to its chances to improve the performance of our *VKB* and *VESA* concepts.

References

Jantke, K.P. et al. 2003. Data Mining Tutor. e–learning system available at *http:/neumann.dfki.uni-sb.de/damit*

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, 2002.

Knauf, R.; Tsuruta, S.; Gonzalez, A.J. 2005. Towards Modeling Human Expertise: An Empirical Case Study. *Proc. of* 18^{th} *International Florida Artificial Intelligence Research Society Conference 2005* (FLAIRS-2005), Clearwater Beach, FL, USA, pp. 232-237, ISBN 1-57735-200-9, Menlo Park, CA: AAAI Press, 2005.

Knauf, R.; Tsuruta, S.; Gonzalez, A.J. 2005. Towards Reducing Human Involvement in Validation of Knowledge-Based Systems. *Proc. of the XIIIth Conference on Knowledge Acquisition and Management (KAM 2005),* Karpacz, Poland, Scientific Papers of Wroclaw University of Economics, No. 1064: *Knowledge Acquisition and Management*, pp. 108-112, ISSN 0342-8445, Wroclaw, Poland: Publishing Co. Wroclaw University of Economics, 2005.

Knauf, R.; Tsuruta,S.; Ihara,H.; Gonzalez, A.J.; Kurbad,T. 2004a. Improving AI Systems' Dependability by Utilizing Historical Knowledge. *Proc. of 10th International Symposium Pacific Rim Dependable Computing*, Papeete, Tahiti, French Polynesia, pp. 343-352, IEEE Computer Society Press, March 2004.

Kurbad, T. 2003. A Concept to Apply a Turing Test Technology for System Validation that Utilizes External Validation Knowledge. Diploma Thesis, Ilmenau Technical University, Dept. of Computer Science and Automation, 2004.

Singh, S. 1998. Fuzzy Nearest Neighbour Method for Time-Series Forecasting. *Proc. of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT'98),* Aachen, Germany, vol. 3, pp. 1901–1905, 1998.

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.