Knauf, Rainer; Tsuruta, Setsuo:

Refinement of validation experience models

Publikation entstand im Rahmen der Veranstaltung:
4th International Multiconference on Computer Science and Information Technology (CSIT 2006), 5. – 7. April 2006, Amman, Jordanien

Refinement of Validation Experience Models

Rainer Knauf
Technical University Ilmenau, Faculty of Computer Science and Automation
rainer.knauf@tu-ilmenau.de, jantke@fit-leipzig.de

Setsuo Tsuruta
Tokyo Denki Unversity, School of Information Environment
tsuruta@sie.dendai.ac.jp

ABSTRACT

The ultimate source of utilizable knowledge seems to be the human mind. However, humans have different beliefs, experiences and learning capabilities. They are not free of mistakes. Humans' opinions about a desired system's behavior differ from each other change over time as a result of misinterpretations, mistakes or new insights. This demands appropriate concepts to support the evolution of their knowledge. The subject of the present paper is validation knowledge, which is used for intelligent systems' evaluation. As models of collective and individual human expertise the authors developed a Validation Knowledge Base (*VKB*) and so called Validation Expert Software Agents (*VESA*s). Both concepts aim at using collective (*VKB*) and individual (*VESA*) experience gained in former validation sessions. Initially, both concepts are designed as subsidiary knowledge to the focal expert knowledge in the validation process, but they have the potential become focal knowledge, which is on a par with original human knowledge. However, a drawback of the models so far was their disability to provide a reply to cases, which have never been considered by any human expert before. In this paper, these concepts are refined by a method to derive answers to questions, which have never been asked before, but which are very likely a commonly accepted answer (*VKB*) within an expert community respectively an individual answer (*VESA*) of a particular expert.

Key Words: Human knowledge models, Collective and individual validation knowledge, Knowledge refinement and revision, Evolution of knowledge models

1. Introduction

The validation process of complex systems requires heavy human participation. 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. The results of these tests are nearly always evaluated by experts. To ensure anonymity and to avoid prejudices, these tests are usually performed by a derivation of the TURING Test [Turing 1950].

However, experts may not always agree among themselves. The number of test cases and the number of experts required for each such exercise can become a great burden 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 Agents* (*VESA*) were developed as a model of individual human expertise [Tsuruta et.al. 2002, Knauf et al. 2004]. These concepts have been implemented in a validation framework [Knauf et al. 2002].

Initially, both concepts are designed to supplement the original human knowledge of the experts who are involved in the validation process. However, the more they exhaust their sources in terms of adopting their knowledge, the more they have the potential to become a focal source of knowledge. In fact, both concepts can be considered as a very fist step towards supporting knowledge evolution:

The *VKB* is collection of most heavily accepted knowledge within an expert community. The idea to identify a community's acceptance is not only to let them rate the solutions of different sources (humans and intelligent systems), but also to weight the rating of a particular human source with an estimated degree of competence for the considered problem case.

The *VESA* is a software agent corresponding to a specific human expert. Its original intention is modeling the validation knowledge of its human counterpart by analyzing similarities with the responses of other experts. It can model individual validation knowledge that is different from the knowledge of the collective majority of experts. Thus, the *VESA* has the potential to maintain excellent and innovative individual human expertise. The main assumption behind the *VESA* concept is that experts who often agree with each other are assumed to agree again when asked for a solution to a new problem case.

Since humans are usually not aware of such similarities, they do not really apply them consciously. Moreover, human brain is not perfect with respect to following their own principles and "thinking structures". Indeed, such accidental "mistakes" are a valuable source of knowledge (like mutations are a valuable source of adaptation). So it might happen (and might even be desirable) that a *VESA* does not exactly model its human origin's validation knowledge (or better: the human origin's answers to questions that require knowledge processing).

Whereas a *VKB* follows a conservative strategy to derive answers to such questions, a *VESA* has the potential to be creative and to come up with answers, which are different from any human answer and which might be more useful than it. Of course, the opposite may also happen, i.e. their answers may be worse then any human's answer, but this is certainly the very nature of evolution. Furthermore, the influence of the *VESA*'s problem solving contribution in the validation process is still limited by a quite conservative counterpart, the *VKB*.

Both concepts were intended as subsidiary knowledge to the focal expert knowledge in the validation process and to assist (next to their human "colleagues") in the complex task to estimate an intelligent system's validity. However, they have the potential to evolve towards focal knowledge on a par with original human knowledge (see [Kaschek et al. 2006] for a discussion of this issue).

To estimate the usefulness of the *VKB* and *VESA* concepts and to reveal their weaknesses, a prototype test was performed [Knauf et al. 2005a]. This test revealed a basic disadvantage of these models. Since they both hack back to authentic human knowledge of former validation sessions, they were not capable to provide an appropriate reply to requests that never appeared in the past.

Although in "toy applications" with a manageable amount of test cases (like the one in [Knauf et al. 2005a]), 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 an identic one has been processed before.

So we looked for an approach to derive answers to those questions, which have never been answered by any *VESA*'s human origin. In fact, an approach of this kind has to be creative and thus, it supports knowledge evolution.

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 to bring about a decision among the k most similar cases in the case base.

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 adaptation towards its use for the intended purpose. In section five we discuss requirements to a test scenario for the suggested approach and section six summarizes the results.

2. The Turing Test Technology

The validation framework introduced in [Knauf et al. 2002] consists of five steps, which can be performed in cycles (see figure 1):

Step # 1: Test case generation Here, an appropriate set of test cases is generated. This set meets the competing requirements (a) *Coverage* of all possible combinations of inputs, which expands the number of test cases to ensure completeness in coverage, and (b) *efficiency*, which limits the number of test cases to make the process practical. This step is performed in two sub–steps: (1) First, a *quasi–exhaustive set of test cases* ($QuEST$) is computed by analyzing the rules and their input/output behavior. (2) Second, the large amount of test cases is limited by utilizing so–called validation criteria. Test cases that don't reach a certain validation necessity degree will be removed from $QuEST$ resulting in a *reasonably sized set of test cases* $ReST$. A workable compromise between these constraints is central to both the technique developed so far and the improvements reached by introducing the VKB.

Step # 2: Test case experimentation Intelligent systems emulate human expertise. Therefore, human opinion needs to be considered when evaluating the correctness of a system's response. Through a TURING Test – like validation approach, this step performs a fair evaluation of the correctness of a system's output by imperfect human expertise. It consists of (1) exercising the set of test data by both the intelligent system and the validating experts and (2) presenting all results – those provided by the system as well as those provided by the human experts – to the validation panel anonymously.

Step # 3: Evaluation The third step interprets the results of the experimentation and determines errors attributed to the system and reports it informally. As a side effect of the previous step, a test case competence assessment of the validators for each particular test case is computed and utilized for a more objective validity statement in the following step by weighting the particular experts' ratings with their respective competence degrees. The competence estimation is based on three sources:

1. a explicitly expressed self–estimation of the expert,
2. an implicitly expressed self–estimation by analyzing his rating of his (anonymously presented) own answer while seeing other experts' answers, and
3. the rating of other experts to his answer to the considered case.

Step # 4: Validity assessment In this step, the results of the evaluation are analyzed and conclusions about the system's validity are drawn. Depending on the purpose of the validation statement, the validity is expressed as (1) validity degrees associated to test cases, (2) validity degrees associated to the system's outputs, (3) validity degrees associated to system's rules, and finally (4) as a validity degree associated to the entire system.

Step # 5: System refinement At the first view, the objective of validation is to gain reliable statements on the usefulness and dependability of an intelligent system. In the

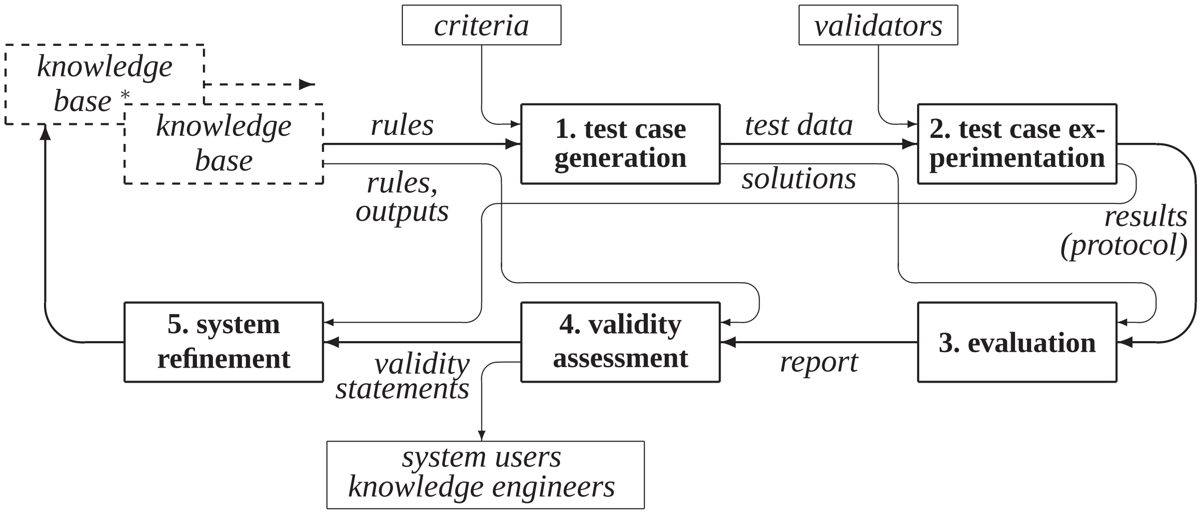

Figure 1: Steps in the Proposed Validation Process [Knauf et al. 2002]

end, however, we are also interested in developing a more dependable system with a better performance. Therefore, this fifth step, which completes the framework, provides guidance on how to correct or decrease the effects of errors or vulnerabilities detected in the system as a result of the previous four steps. Since the validity assessment points out the rules which infer invalid solutions and the TURING Test experimentation reveals a so–called optimal solution to each test case, we are able to refine these rules with the objective to provide the optimal (i.e. most dependable) solution. This, naturally, leads to an improved input–output behavior of the system, and thus, to a more dependable system.

The benefit of this standardized validation framework is that developers of knowledge–based systems can reference it when describing the validation process to the end user. This may enhance the acceptance of the system. Furthermore, this framework attempts to minimize the effort involved in validation of the expert system. This is because cases derived from the knowledge in the VKB don't have to be resolved in the process. The reason not to resolve them is that the VKB is intended to serve as a source of **external** knowledge, which consists of a historical solution that obtained good marks in the past. Lastly, this minimized effort leads to reduced and more predictable costs. A more comprehensive description of all steps as well as the research behind this work can be found in [Knauf et al. 2002], e.g.

3. The Concepts of VKB and VESA so far

Due to the heavy involvement of humans, the expensive expensive step of our 5–step validation framework [Knauf et al. 2002] is the test case experimentation step. The recently proposed concepts of *VKB* and *VESA* aim at reducing this cost factor significantly [Knauf et al. 2005b].

The *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. 2005a].

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_{Kj}^{opt} 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, 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 2 sketches how the VKB is employed in the test case experimentation.

A *VESA* is requested, in case an expert e_i is not available to solve or rate a case t_j. So far, e_i's or similar e_j's former (latest) solution is considered by this expert's *VESA*. In fact, it can be considered as its human origin's model. Its answers to questions require a knowledge analysis and knowledge processing and might differ from the human origin's answer. So these models have properties that their origins don't have and thus, they follow very much the understanding of the term model as a result of a nice discussion held in [Kaschek et al. 2006].

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. 2005a].

Formally, *VESA* acts as follows, when a *VESA*$_i$ is asked to provide a solution for a test case input t_j on behalf of its expert e_i:

1. In case e_i solved t_j in a prior validation exercise (with a value other than unknown), his/her solution with the latest time stamp τ will be furnished by VESA$_i$.
2. Otherwise, *VESA*$_i$ performs the following steps:

   (a) All validation experts e', who ever delivered a solution to t_j form a set $Solver_i^0$, which is an initial dynamic agent for e_i: $Solver_i^0 := \{e' : [t_j, E_{Kj}, \ldots] \in VKB, e' \in E_{Kj}\}$

   (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 sol_{Kj}^{opt} and in the same session τ. e_{sim} forms a refined dynamic agent $Solver_i^1$ for e_i: $Solver_i^1 := e_{sim} : e_{sim} \in Solver_i^0, |\{[t_j, E_{Kj}, \_, sol_{Kj}^{opt}, \_, \_, \tau, \_] : e_i \in E_{Kj}, e_{sim} \in E_{Kj}\}| \longrightarrow max!$

   (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 by VESA$_i$.
3. If there is no such most similar expert, VESA$_i$ provides $sol := unknown$.

If a VESA$_i$is requested to provide a rating to a solution of a test case input t_j on behalf of expert e_i, it models the rating behavior of e_i as follows:

1. In case e_i rated t_j in a former session, VESA$_i$ adopts the rating with the latest time stamp τ and provide the same rating r and the same certainty c.
2. Otherwise, *VESA*$_i$ performs the following steps:

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

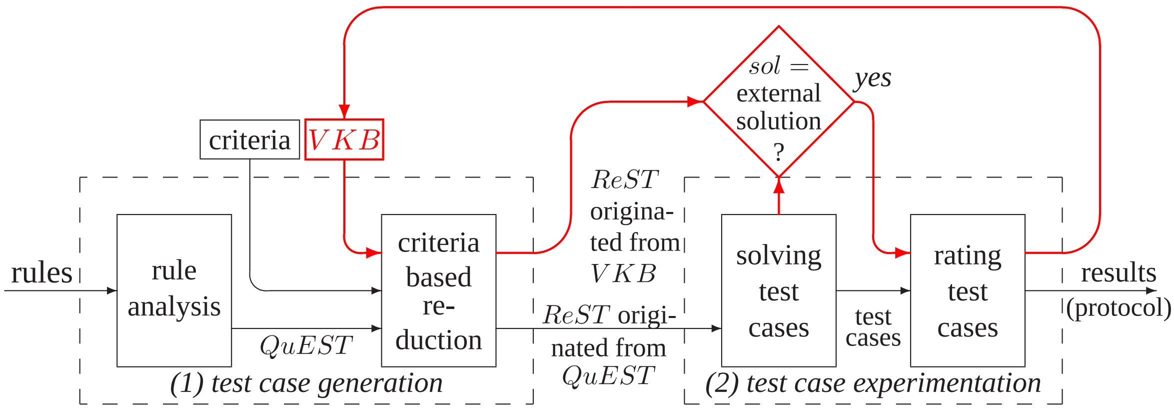

Figure 2: The use of the VKB in the Test Case Generation and Experimentation

(a) All experts e', who ever delivered a rating to t_j form a set $Rater^0_i$, which is an initial dynamic agent for e_i: $Rater^0_i := \{e' : [t_j, \_, E_{Ij}, \ldots] \in VKB, e' \in E_{Ij}\}$

(b) Select the most similar expert e_{sim} with the largest set of solutions sol^{opt}_{Kj} that have been rated by both e_i and e_{sim} with the same rating r_{IjK} and in the same session τ. e_{sim} forms a refined dynamic agent $Rater^1_i$ for e_i: $Rater^1_i := e_{sim} : e_{sim} \in Rater^0_i, |\{[t_j, \_, E_{Ij}, sol^{opt}_{Kj}, r_{IjK}, \_, \tau, \_] : e_i \in E_{Ij}, e_{sim} \in E_{Ij}\}| \longrightarrow max!$

(c) VESA$_i$ provides the latest rating r along with its certainty c to the present test case input t_j of e_{sim}.

3. If there is no such most similar expert e_{sim}, VESA$_i$ provides $r := norating$ along with a certainty $c := 0$.

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 counterpart.

EK_3	solution of *VESA*$_2$	solution of e_2	EK_3	solution of *VESA*$_2$	solution of 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

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").

Both concepts *VKB* and *VESA* as developed so far, rely on the availability of an entry $[t_j, \_, \_, \_, \_, \_, \_, \_]$ in the *VKB*, when they are asked for a solution or rating to a test data t_j. If nobody considered t_j in any pre-

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

vious 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.

4. 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 $f : \mathbb{R}^n \rightarrow V$[1]. Objects along with a known function value form a set of *examples*.

A *similarity* (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:

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

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

with

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

Figure 3 shows a two–dimensional example with $V = \{\oplus, \otimes\}$. Here, different values of k result in different class memberships 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

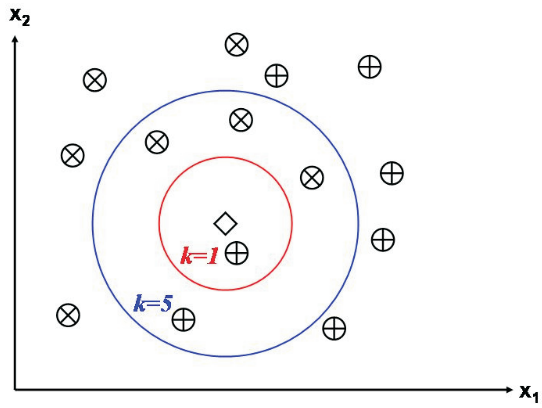

Figure 3: The influence of the parameter k

that the method 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:

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

with

- $\delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases}$
- $\omega_p = \dfrac{1}{d(y, x_p)^2}$

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.

5. Adapting the Method towards a Knowledge Refinement Concept

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 (the class membership) we look for. The data is a vector of *test data components* $t_j = [s_j^1, s_j^2, \ldots, s_j^p]$. An example is formed by the respective components along with the (known) class membership sol_{Kj}^{opt} and its 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 function vales, 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 a *VKB* has to meet the following requirements:

1. Each test data component should have the chance to 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.
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 of t_j as well the data of the case to be classified $t_{j\diamond}$ in a way, that each component is finally 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 $\hat{s}_j^i = \dfrac{n_j}{max}$ respectively $\hat{\tau} = \dfrac{\tau}{max}$
- 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

[2] In the Data Mining Tutor at

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 pre-processed 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.

5.1. 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:

$$sol(t_j) = \begin{cases} sol^{opt}_{K_J j} & \text{if } E \\ \max\limits_{\hat{t_i} \in T} \sum\limits_{i=1}^{7} \omega_p * \delta(sol_j, sol_i) & \text{else} \end{cases}$$

with

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

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, 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.

5.2. Refining the VESA concept

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 most similar to t_j: $T = \{\{\hat{t_1}, \ldots, \hat{t_7}\} : \neg\exists\hat{t^*} : d(\hat{t_j}, \hat{t^*}) \leq \max\limits_{i=1,\ldots,7}(d(\hat{t_j}, \hat{t_i})\}$.

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 $Solver^0_i$, which is an initial dynamic agent for e_i: $Solver^0_i := \{e' : [t_k, E_K, \ldots] \in VKB, t_k \in T, e' \in E_K\}$.

2. Select the most similar expert e_{sim} with the largest set of cases that have been solved by both e_i and e_{sim} with the same solution sol^{opt}_{Kj} and in the same session τ. e_{sim} forms a refined dynamic agent $Solver^1_i$ for e_i: $Solver^1_i := e_{sim} : e_{sim} \in Solver^0_i, |\{[\_, E_K, \_, sol^{opt}_{Kj}, \_, \_, \tau, \_] : e_i \in E_K, e_{sim} \in E_K\}| \longrightarrow max!$

3. Determine the set $VKB(e_{sim}, T) \subseteq VKB$ 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 $VKB(e_{sim}, T) \subseteq VKB$.

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 $k = 7$.

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 solutions sol_{Kj}^{opt} that have been rated by both e_i and e_{sim} with the same rating r_{IjK} and in the same session τ. 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 $VKB(e_{sim}, T) \subseteq VKB$ of ratings to any case $t \in T$, which are provided by e_{sim}: $VKB(e_{sim}, T) = \{[t, \_, E_I, \ldots] : t \in T, e_{sim} \in E_I\}$.

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 $s^1, \ldots, s^p$ 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 zero ($c := 0$).

5. If the *VKB* is too small to determine T, *VESA*$_i$ provides $r := norating$ along with a certainty $c := 0$.

6. Summary

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

A *VKB* and the *VESA*s are conservative (*VKB*) and creative (*VESA*) approaches to model human validation knowledge respectively to simulate the result of human knowledge analysis and processing. At least the latter approach has the potential to behave different from its human origin when requested to provide solutions or validity statements to particular cases. More generally, they may be considered as a host in which "mutations" of the original human knowledge are constructed. In fact mutations are the natural source of evolution.

The capability of both models to provide answers to questions that come up in the validation process was quite limited so far. These models suffered from not providing a requested reply to cases that have never been considered by human expert panels in the past.

To complete the performance of these models towards providing solutions or validity estimations to any case, 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 a solution to a new case, the solutions 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, we introduced a method to pre–process the examples cases in the *VKB* for using the k–NN method.

Our upcoming research on this approach faces three issues: (1) an empirical evaluation of the approach by a prototype experiment, (2) a derivation of an appropriate k for successfully applying the k–NN method, and (3) 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.

In fact, (1) is an ambitious task. Generally, the method that has been used in former experiments [Knauf et al. 2005a], can be utilized again. A *VKB* can be validated by considering the ratings of the human experts to a solution that has been proposed by the *VKB*. Since the refined *VKB* now always delivers solutions, one part of the experiment can even be omitted. A *VESA* can be validated by comparing its results with the ones of its human counterpart. On the one hand this is the only quality measure we have (so far), on the other hand, it might be desirable in to come up with mutations. So we tend to include a consideration of the ratings received for these mutations.

References

[Kaschek et al. 2006] Kaschek, R.H.; Jantke, K.P.; Nebel, I.-T. 2006. Towards Understanding Meme Media Knowledge Evolution. Jantke, Lunzer, Spyratos, Tanaka (eds): *Federation over the Web*. International Workshop, Dagstuhl Castle, Germany, 2005, Proceedings, volume 3847 of Lecture Notes in Artifical Intelligence, pp. 183-201, Berlin, Heidelberg, New York: Springer–Verlag, 2006.

[Knauf et al. 2002] 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 et al. 2005a] Knauf, R.; Tsuruta, S.; Gonzalez, A.J.: Towards Modeling Human Expertise: An Empirical Case Study. *Proc. of* 18^{th} *International Florida Artificial Intelligence Research Society Conference 2005* (FLAIRS-2005), pp. 232-237, ISBN 1-57735-200-9, Menlo Park, CA: AAAI Press, 2005.

[Knauf et al. 2005b] Knauf, R.; Tsuruta, S.; Gonzalez, A.J.: Towards Reducing Human Involvement in Validation of Knowledge-Based Ssystems. *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, 2005.

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

[Kurbad 2003] 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.

[Tsuruta et.al. 2002] 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)*, pp. 226–230, Menlo Park, CA:AAAI Press.

[Turing 1950] Turing, A.M. Computing machinery and intelligence. *Mind*, LIX, number 236, pp. 433–460, 1950.