Formal Aspects of Case-Based Decisions Making Support by Wells Drilling

Vasyl Sheketa[0000-0002-1318-4895], Mykola Chesanovskyy[0000-0002-8687-0837]

Volodymyr Pikh[0000-0001-9420-5522], Yulia Romanyshyn[0000-0001-7231-8040],

Lyubov Samaniv[0000 0003 1693 1961] , Mykola Pasyeka[0000-0002-3058-6650]

Ivano-Frankivsk National Technical University of Oil and Gas,
vasylsheketa@gmail.com, nick.chesanovskyy@gmail.com,
pixclg@gmail.com, yulromanyshyn@gmail.com,
lmsamaniv@gmail.com, pms.mykola@gmail.com

Abstract. The outlined concept of case-based drilling data contains expert experience in the subject area and satisfies the criteria of the main sources of knowledge required for the functioning of case-based reasonings as such. Among the main sources are: case database, similarity metrics and adaptation containers in the form of constraints systems. In the general case, it is stated that *case-based decision support* technology performs the ranking of operator actions to establish the values of controlled parameters, and the predefined goals can be interpreted as the solution of relevant technological problems in the imposed systems of constraints. Existing cases, thus, serve as guidelines for advancing of the solution to increase its relevance and allow effective adaptation of existing solutions, their parts and generalizations in the form of templates and samples to the given new conditions, including those in the form of imposed constraints system, where solutions will be considered correct if they satisfy the imposed constraints in the form of technological regulations in full or at least partially.

Keywords: case-based, constraints, reasoning, decisions-making, intelligent decisions-making support systems, machine learning.

1. Introduction

When it comes to the formal foundations of case-based modelling and case-based solutions[1–5], there primarily is all about the classical mathematical characteristics, such as: correctness, completeness and complexity of the created systems. At the implementation stage, it is important that the behavior of the system should be as predictable as possible, so a formal description of the solution is important, above all, in terms of the possibility of its successful verification, including in the form of a software product. The explored methodology itself is not continuous and is divided into a number of stages, as well as from a point of view of formal justification it cannot be continuous as well by definition and, accordingly should consists of a number of formal theories, which, in fact, are relevant in this application. Starting with the theory of logic, the sequence of constructed solutions of problems is most suitable for describing the way of presenting knowledge for decision support. The most convenient will be description by means of probabilistic approaches for *case-based decision support*. Namely, descriptive logic can be considered as a formalism of representation of knowledge-oriented entities to the subject area, which basically contains the corresponding taxonomy for complex objects. The application of such formalisms is to present case-based indices as concepts in the descriptive logic. Thus, the specifications of problem descriptions should be reflected in the corresponding indexes, which allows the implementation of case extraction based on reasoning in terms of approximation and similarity. In particular, in the applications of *intelligent decision support systems* [6–10] precedent(case)-based reasoning methodologies are actually a way to effectively incorporate expertise into decision-making. It is from this point of view that this methodology is an integral methodology for the processes of building knowledge-oriented systems. This is a way to effectively adapt past expertise to solve new problems. In turn, the solution of new problems allows to generate some experience, which will strengthen the intelligent systems. Thus, in the initial approximation, the essence of the explored methodology can be reduced to *machine learning* in the process of solving new problems. So far, in order to solve certain problem we need some minimum set of knowledge (volume of knowledge in the knowledge base) at the same time. Having solved a certain problem, we do update the knowledgebase by making new consistent occurrences in the form of knowledge entities, which expands the entire scope and ability of the system to solve relevant new problems or even whole classes of problems. Therefore, by analogy with human experts, it can be stated that the system must work out a certain or some basic number of case studies, which can be considered as basic, respectively, to obtain some minimum level of "skills" in terms of human experts. Thus, for each subject area, respectively, we can identify some basic sets of case studies, which can also be interpreted as typical, most common i.e. At the level of the knowledge base for the case database, we will receive some core of the knowledge base $Core(KB)$, which can form its guide set of inertia in the process of *expected modifications*. An important fact, clear from the general theory of knowledge bases, is that we can make a $Core(KB)$ entry immediately without the need to train the system on these case studies. And the case-entry will be immediately guaranteed to be true. From the point of view of artificial intelligence, it will also be important to note that such an initial initialization of the system in the form of $Core(KB)$ will set the basic gradient and the corresponding scenarios of its reasonable behavior in solving problems in the subject area. Forming in this sequence and this way the construction process of *knowledge-oriented system* [11–12], it should be borne in mind that the structure of its knowledge should not only describe the *signature* of the relevant cases (*cases signature*), but also describe the processes of displaying elements of these signatures that do not correspond to the concept of knowledge as a whole, to perform calculations of the corresponding similarity levels to determine the appropriate ways to adapt the correct and satisfactory solutions to the selected problems by their possible *modification* at the level of entry parameters and their ranged values what defines the main scope and aim of the *proposed research.*

2 Formation of case-based decision-making support guidelines

Let consider the question of constructing of signature for a typical case in the subject area of drilling oil and gas wells [13–14] in the context of *decision-making support by intelligent system*. We will start considering the sequence of controlled parameters *(tcp* - parameters), unmanaged parameters (*ucp* - parameters), disturbing parameters (*dcp* – parameters) and the output resulting parameters (*ocp* - parameters). Based on the given problem (technological problem in the field of drilling of oil and gas wells), the methodology of reasoning should be applied accordingly in order to extract past similar cases for the purpose of repeated or modified application for problem solving in the process of decision-making support: $Sol(TP)$, what is in them $Sol(TP)|Case_i$ or $Sol(TP)^{mdf}|Case_j$ where $i,j \in N$.

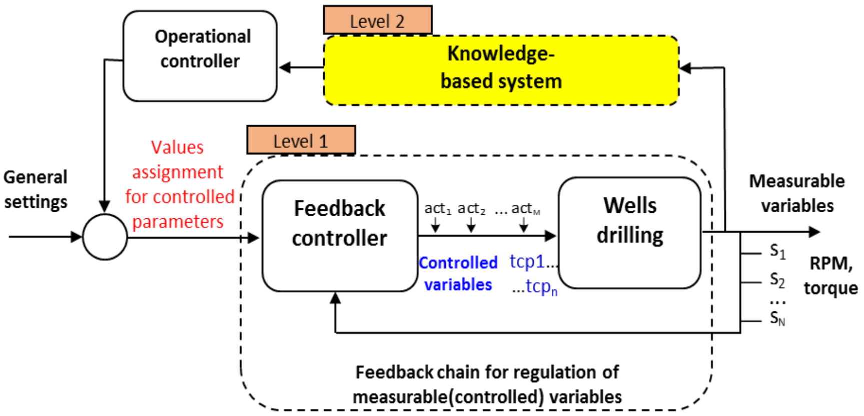

Fig.1. Scheme of intellectual control

where $act_1, act_2, \ldots$ the sequence of actuators installed, $s_1, s_2, \ldots$ – a sequence of sensors to obtain the actual values of the parameters.

Thus, at all stages of operation the system is based on a certain amount of knowledge, which, as noted above, is nothing more than a common domain of knowledge and in the form of problems (case studies) and methods of solving them in the form of their own solution: $Sol(Case)$. Actually, after the problem is posed and formed, it can already be considered as a certain state in the space of solutions of all possible problems. Solutions space, provided there is some certain and unknown solutions and the correct solution of each problem, which will be to move around the space of stands of the desired relevant solution in the form of the required substitution of values of controlled technological parameters.

The projected methodology can be considered from the Bayesian approach[15], where attributes (parameters) are interpreted as corresponding random variables, namely the case representation (or case-like representation) is used to approximate the combined probability of the entered attributes (parameters). Thus, this approach will require the introduction of sequences of values of discrete attributes (parameters). Further development of such a representation, combining probabilities will lead to the necessary consideration of the set of cases, and the desired approximation will be achieved by clustering data attributes (parameters) by grouping the corresponding cases that have similar properties. Bayesian models in our case will be determined by a set of random variables (parameters) $\langle tp_1,\cdots tp_n \rangle$. Accordingly, some case $\overrightarrow{case_i}$ will be an instance in form of substitution of values for the variables (parameters): $\overrightarrow{case_i} = \left(\underbrace{tp_1 = v_1,\cdots,tp_n = v_n} \right)$. Then accordingly to the introduced presentation way, the database of cases $Cbase$ will be the set of m independent and identical distributed datasets in the form of substitutions. In the next step, the set of cases can be clustered into L groups and the corresponding probability distribution represented by such a cluster will be obtained. Then we will have that for every L $prb\left(\overrightarrow{Case_i}\right)\left|X = X_l\right|$ will express the probability that the case will belong to cluster X_l , where X is a random variable correlated with clusters. Thus, in the next step it can be argued that each case can be approximated by the weight sum of the corresponding distributions of the form $prb\left(\overrightarrow{Case_i}\right) = \sum_{l=1}^{L} prb\left|X = X_l\right| prb\left(\overrightarrow{Case_i}\left|X = X_l\right|\right)$. Assumed that the parameters tp_i within each cluster are independent, we will have that $prb\left(\overrightarrow{Case_i}\right) = prb\left(tp_1 = v_1,\cdots,tp_n = v_n\right) = \sum_{l=1}^{L} prb\left(X = X_l\right) \prod_{i=1}^{n} prb\left(tp_i = v_i \left|X = X_l\right|\right)$. On the basis of such model the decision of various type of probability-based reasonings actually at construction of the decision type becomes possible: $Sol(tcp)^{prb}$. Thus, after performing a case database check $Cbase$ and instantiation $\overrightarrow{Case_{Qm}}$ as a *query case*, we will be able on the basis of the above to define the predictive distribution as $prb\left(\overrightarrow{Case_{Qm}}\left|Cbase\right.\right) = prb\left(\overrightarrow{Case_{Qm}}\left|Cbase,\Delta\right.\right)$, where Δ indicates the corresponding model parameters $\Delta = (cd,tcp) = (\delta^{cd},\delta^{tcp})$, where δ^{cd} - parameter that describe the cluster distribution $\delta^{cd} = \left(cd_1,\cdots,cd_l \left|cd_i = prb(X = x_i)\right.\right)$, and δ^{tcp} - parameter related to the conditional probabilities of clusters in relation to the values of variables $\delta^{tcp} = \left\{tcp_{ij}\right\}$, where each $\left\{tcp_{ij}\right\}$ is a set of *controlled parameters*. For discrete variables with a certain cardinality $Card\left(D_i,tcp_i\right)$ in CSP – notation respectively, we will get that

$$tcp_{ij} = (tcp_{l_1},\cdots,tcp_n,\cdots,tcp_{l_1},\cdots,tcp_{l_n}) \simeq Card(D_i,tcp_i) \simeq prb(tcp_i = v_l \left|X = x_j\right|) .$$

Given the initial assumption that the cases in $Cbase$ are expectedly independent at a given Δ , we will have that $prb\left(\overrightarrow{Case_{Qm}}\middle|Cbase\right) = prb\left(Case_{Qm}\middle|\Delta\right)$.

From the point of view of the real technological process of wells drilling, which takes place in conditions of *uncertainty*, it is necessary to determine the values of non-instantiated variables tp_i , which are the solution of a search problem for a *case query* at some given instantiated values that form a description of the problem

$$\boldsymbol{Sol}(TP \mid= Q_m) = UnInst(\cdots tp_i \cdots) \middle| Inst(\cdots tcp_{i-2} \cdots) .$$

Accordingly to the classical approach, assume that n-1 of the first parameters $tcp_i : tcp_1,\cdots,tcp_{n-1}$ are appropriately instantiated and take values $v_1,\cdots,v_{n-1}$ in accordance. Then in terms of the introduced representations it is necessary to define distribution $prb\left(tp_n \middle| Cbase,\left[tcp_1,...,tcp_{\boldsymbol{n-1}}\right]\right)$ based on the essence of the specified parameters of the model

$$prb\left(tp_n = v_n^i \middle| \Delta, tcp_1,\cdots,tcp_{n-1}\right) = \frac{prb\left(tp_n = v_n^i, tcp_1,\cdots,tcp_{n-1} \middle| \Delta\right)}{prb\left(tcp_1,\cdots,tcp_{n-1} \middle| \Delta\right)} .$$

We determine the recalculation of this formula in terms of the given instances

$$\underline{\sum\nolimits_l^L prb}\left(X = x_l \middle| \Delta \middle| prb(tp_n = v_n^i \middle| X = x_l, \Delta)\, \Pi_{k=1}^{n-1} prb(tcp_k = v_k \middle| X = x_l, \Delta)\right) .$$

This representation is the basis of the model that can be used to perform the classification in terms of constructing the division into discrete tcp_i - classes. For this purpose, the above expression will take the form

$$\underline{\sum\nolimits_{l=1}^L prb}\left(X = x_l \middle| \Delta\right) \Pi_{k=1}^{n-1} prb(tcp_k = v_k \middle| X = x_l, \Delta) .$$

This type of representation allows, respectively, among all possible values for tcp_i to select the most probable values insofar.

Fuzzy sets[16] can be thought of as a set of objects with a continuing sequence of degrees of membership. Therefore there is need to consider the concepts based on the principle of the attributes similarity. In our case, the attributes are drilling parameters. So, the fuzzy model will be based on the degree to which the previous case belongs to a set of sufficiently similar cases in relation to the current problem. According to this principle, the whole process of reasoning will be based on the appropriate level of similarity, as opposed to the level of instances, and is directly related to the definition of the level of similarity(sd) and the way it is measured for the problem description space: technological process states $TP.State$ and solution space $\boldsymbol{Sol}.Space$ accordingly. It is important that $TP.State$ i $Sol.Space$ both are fuzzy relations defined in the range $[0,1]$ respectively, and are applicable to pairs $\left\langle atribute\left(parameter\right) \text{-} value\right\rangle$, as a type of corresponding representation on a *set of cases*:

$$\forall\left(tp_i, Sol_i\right),\left(tp_j, Sol_j\right) \in Cbase, \boldsymbol{Sol}.Space\left(tp_i, tp_j\right) \le TP.State.Space\left(sd_i, Sol_i, Sol_j\right) .$$

This representation means that the similarity ratio for the space of the technological problems limits the similarity ratio in the solution space. Thus, for example, if two problems are similar, then their solutions should be as similar as the corresponding problem descriptions. Thus, by solving a new problem $Case_{Query} = \left\langle TP_{Query}, Sol_{Query}\right\rangle$, where the solution is initially unknown, it is the constraints imposed that will determine the set of possible values for Sol_{Query}, namely:

$$Sol_{Query} = \cap \overbrace{(TP,Sol)}^{Case} \in Cbase\left\{ Sol_{Query} \in TP \middle| Sol.Space\left(tcp_i, tcp_{Qm}\right) \leq TP(Sol_i, Sol_{Qm}) \right\}.$$

So, in general, probabilistic formalization [17] is based on the same approach as the fuzzy formalization mentioned above. Namely, the solution space is limited by similarities at the level of the problem description space $TP.\left[State,Space\right]$, then probabilistic models allow reasoning also at similarity levels.

In turn, respectively, both types of models - *fuzzy and probabilistic* allow the processing of incomplete and uncertain information, which is considered as a kind of approximation of reasoning, as both are based on the hypothesis that "similarities at the input respectively give similarities at the output (as a result) »(as can be seen at Figure 1), therefore, in both cases of formalization we can speak of a case-based inference (precedent-based inference), and in the case of probabilistic representations it is a question of probability distribution based on similarities at the input.

Thus, case-based inference takes place at the level of similarities, where there is a certain type of reflection from the initial instance level of *controlled parameters values substitution* to the corresponding level of similarities. In summary, the sequence of the main steps in terms of probability-based approach can be reduced to: 1) characterize the problem at the level of similarities by means of the existing structure of similarity between parameters; 2) use the obtained similarity scheme in order to derive probabilistic characteristics for unknown resulted parameters; 3) do perform the translation of similarity levels for results at the level of their instantiation.

Thus, the step of obtaining will be to modify the system by correcting failures, improving performance or other system properties, respectively, or adapting the system to changes in the environment by bringing it to a state known as correct, perfect with adaptive mode. One of the important features is that this stage is integrated into the methodology itself. Thus, in the execution mode, the system constantly collects the relevant data from the case-based system and redirects the maintenance operation accordingly to the specified scenario(*drilling mode* for example : *optimal, rational, forced* etc.). It includes distinct and different stages: retention, review and recovery. Content includes proper support, such as drilling process failure handling(*state of emergencies*) and problem solving. Moreover, the problem of *modifying knowledge* within the system when deciding whether or not to add new solved cases to the database and their sequences, respectively, for processing in the task plan of generalized case sequences. At the same time, this stage performs the actual type of *decision support*, which can ultimately lead to a significant reduction in the effectiveness of solving the problem in accordance with the actual size of the case database or in accordance with the accumulation of complexity through not distributed collections of cases or other complexities related to the complex nature of real-world subject area data. Thus, the review and recovery will provide the mechanism for managing support over a much wider range, taking into account the required level of accuracy and quality of measurement in the system in relation to the existing knowledge containers.

The simplest systems, respectively, contain only the retention stage. Full-featured systems must also contain stages of review and recovery, which is a mandatory task for the solution in the design of modern knowledge-based systems, but their actual implementation depends entirely on the subject area of application. Conceptual fluctuation is a type of *machine learning* problem in which the target concept changes depending on the context. Moreover, the frequency of such changes is different, so accordingly, we can distinguish different degrees of fluctuations. It is also important that the elements of the context that cause such fluctuations are in most cases unknown or hidden. Therefore, systems with a *decision support* mechanism are able to track such changes and, accordingly, to adapt to them as such. Each application domain accordingly includes a specific *support strategy*. The support strategy is accordingly described in terms of collecting relevant data for the support process, how they decide to run *support procedures*, access and type of support operations, and how the selected support operations will be performed accordingly. The next stage starts after evaluating the case solution and its *necessary modifications*, if needed. This stage also launches the support phase after gaining some experience in solving the problem, as a result of revision of the results, which accordingly provides a number of opportunities to implement training procedures, namely:

1) the *stage of learning* – the successful completion of the problem-solving process and the dynamic preservation of a new case in such a way that:
- the structure of the case database can be easily modified;
- you can get generalizations for newly added cases;

2) start the *training procedure* in case the solution of the problem has failed. In this case, additional substages are contained [18]:
- by studying the errors;
- by studying the context relevant to the error.

Accordingly, the decision as to whether or not to store cases depends on whether the system has an appropriate intra-oriented support strategy or not, such as analyzing the state of the case database for future problem solving or the appropriateness of a particular decision as such in general. In the general case, the problem of expediency concerns, first of all, the size of the database of cases, the average extraction time, because if the case is added to the database of cases, a "saturation point" of a certain level can be reached and the efficiency of the system begins to decline significantly. For example, the system that follows the method of outputting answers at the reuse stage may indicate an appropriate expediency metric to control the addition of cases, based on the corresponding computational effort, which, in fact, indicated by the successfully extracted cases and these efforts were transferred to the priority tasks when gaining experience in solving the problem (i.e. *expert experience*).

When a particular class can be stored in memory for later use, it is then that the case database must be *modified* to fully match the particular case. For example, if the database of cases is organized hierarchically, then each new case should be suspended in the corresponding node and the information concerning some node should be updated accordingly. Such a *modification* can be global depending on which *machine learning method* was used. According to the case modification strategy, when the actual case memory includes prototypes, the inclusion of a new case should also indicate the use of some *inductive machine learning method* to obtain a specific prototype, respectively.

Therefore, in this case the wide range of possibilities turns out and some certain class of approaches concerning machine learning too can be chosen, accordingly during initialization of case database.

Consider some specific case $Case_{Qm}$ and assume that it was solved by extracting the case $Case_{mined}$ and the difference between the two cases is accordingly based on the value of the attribute tp_i, so that $TP_l\left(tp_i\right) = V_l$, $TP_{Qm}\left(tp_i\right) = V_{Qm}, V_l \neq Q_m$. Then, accordingly, the prototype TP , representing both types of cases, can be respectively formed on the basis of two attributes (parametric values), namely $TP_{tp}\left(tp_i\right) = \left\{V_l, V_{Qm}\right\}$. Moreover, if tp_i is a taxonomic attribute (parameter), then we also have that $TP_{tp}\left(tp_i\right) = msg\left\{V_l, V_{TP}\right\}$, where $msg(\ )$ is a kind of necessary most specified generalization.

In turn, it is also important that the problem of studying *errors*[18] in practice will mean the need to preserve error cases to avoid providing an incorrect solution for the same description of the problem, respectively. The study of errors also means the revision of all containers of knowledge to identify and analyze those elements of knowledge that led to the general failure and, accordingly apply a *necessary modification*[19] to avoid such a situation in the future.

All modifications should include the actually adjusted cases $Case_{corr}$ with a new correct solutions added. Then, accordingly, the priority task to perform at modification stage will be the outlining of relevant knowledge to distinguish the extracted false cases $Case_{err}$ from the desired cases – $Case_{sol}$. Such a sought-after case $Case_{sol}$ is the closest case to case $Case_{corr}$, which must be extracted in accordance with the final decision $Case_{fin}$. Thus, knowledge about the divergence of cases may include various distinctive features (properties), their descriptions and different types of adaptation rules among the elements of occurrences. Then, the indices are related to $Case_{sol}$ and $Case_{err}$ should be updated after performing the appropriate analysis, namely $Case_{sol}$ should have a better chance of being mined than $Case_{err}$ in equally similar contexts.

Accordingly, each property (feature) $\left[tp_i = V_{Case}\right]$ has some weight value $Weight\left(\left[tp_i = V_{Case}\right]\right)$. Properties are updated depending on whether they add something to correct selected cases from memory. Thus, the following situations can be distinguished: 1) $Case_{\boldsymbol{Err}} = \left(TP_{err,} Sol_{err}.Sol^{err}, Out\right)$; 2) $Case_{sol} = \left(TP_{err,} Sol_{err}.Sol^{err}, Out\right)$, where TP_{err} i TP_{sd} present descriptions of the problem accordingly; 3) if $\left(\left[tp_i = V_{Case}\right] \in TP_{sol}\right)$ i $\left(\left[tp_i = V_{Case}\right] \notin TP_{err}\right)$, then we will have that

$$Weight\left(\left[tp_i = V_{Case}\right]\right) = Weight\left(\left[tp_i = V_{Case}\right]\right) + \boldsymbol{CF}^{\boldsymbol{1}}\, Weight\left(\left[tp_i = V_{Case}\right]\right)$$

where, $\boldsymbol{CF} \in \left[0,1\right]$, which also means that the properties (features) presented in the desired case, but not in the erroneous case (error case), must match the relevance of some higher level, but was not detected during the comparison; 4) if we have the case that $\left(\left[tp_i = V_{Case}\right] \in TP_{err}\right)$ and $\left(\left[tp_i = V_{Case}\right] \in TP_{sol}\right)$, then we will have accordingly that

$$Weight\left(\left[tp_i = V_{Case}\right]\right) = Weight\left(\left[tp_i = V_{Case}\right]\right) - CF^2\ Weight\left(\left[tp_i = V_{Case}\right]\right).$$

This situation means that the properties (features) are presented in the false case, but not in the desired case, did not match accordingly:

$$\textbf{if}\left(Weight\left(\left[tp_i = V_{Case}\right]\right)\right) \in TP_{err} \cap TP_{\boldsymbol{Sol}},\ \textbf{then}\ Weight\left(tp_i = V_{Case}\right) = Weight\left(tp_i = V_{Case}\right).$$

This means that the features (properties) characteristic of both cases have not been modified. In any case, there is no doubt that all these operations are applied to non-empty sets, ensuring appropriate changes in the knowledge base and avoiding appropriate repetitions and failures. On the other hand, a number of possible variations to this situation as such should be considered. It is important, in fact, that in simpler approaches it is necessary to update the features (properties) of relevance in all cases where the cases are successfully extracted, and the values of relevance in all other cases, respectively. At the same time, much more *refinement modifications*[19] must be made taking into account the probabilistic characteristics for each of the classes. The value of $\boldsymbol{CF}$ plays the role of an indicator of the level of learning, so the appearance of large values will mean, respectively, large changes and *modifications* when trying to solve the problem with given deviations of *controlled parameters* and, accordingly, the convergence of values for $Weight\left(\left[tp_i = V_{Case}\right]\right)$ in the presence of such an information need sufficiently small values for $\boldsymbol{CF}$, and will imply the need for modifications.

3 Conclusions

The essence of the use of case-based considerations in solving technological problems of the wells drilling process is defined, which ultimately allows for all the relevant operations to be actually reduced to operating with values sets represented in the form of entities with imposed constraints that can be quantified by both quantity(parameters number) and quality(crispiness) of presentation. Thus, by such a premises, case-based reasoning is one of the effective methodologies for building knowledge-oriented systems for decision making support by wells drilling, where the central element is past expertise of drilling operators in the form of cases (precedents). It is clear that the more such cases, the better for the system, the higher the quality of machine reasoning and *decision making support* respectively. In the context of decision making support is important not so much the very process of reasoning of the system, but the result of such reasoning, which should lead to a solution of the technological problem, which is described by the process of forming of an solution space for selected technological states with imposed constraints. Thus, in the general case, it is also important that the projected probabilistic approach to case-based inference is a significant extension of the most common crispy case of statistical reasoning, because such an extended range of expert assessments and judgments, in fact, does operate accordingly to similarity of the content, in principle, for the identical generations distributions of controlled variables without reference to the essentially subjective evaluations of experts, which is of much more importance insofar when it is about human experience based knowledge .

References

1. Agnar Aamodt and Enric Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," Artificial Intelligence Communications 7 (1994): 1, 39-52.
2. Richter, Michael M.; Weber, Rosina O. (2013). Case-based reasoning: a textbook. Heidelberg: Springer Verlag. doi:10.1007/978-3-642-40167-1. ISBN 9783642401664. OCLC 857646182.
3. Eyke Hüllermeier. Case-Based Approximate Reasoning. Springer-Verlag, Berlin, 2007.
4. Begum, S.; M. U Ahmed; P. Funk; Ning Xiong; M. Folke (July 2011). "Case-Based Reasoning Systems in the Health Sciences: A Survey of Recent Trends and Developments". IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews. 41 (4): 421–434. doi:10.1109/TSMCC.2010.2071862. ISSN 1094-6977
5. Bergmann, R., Althoff, K.-D., Breen, S., Göker, M., Manago, M., Traphöner, R., and Wess, S. *Developing industrial case-based reasoning applications: The INRECA methodology.* Springer LNAI 1612, 2003.
6. Power, D.J. *A Brief History of Decision Support Systems.* DSSResources.COM, World Wide Web, http://DSSResources.COM/history/dsshistory.html, version 4.0, March 10, 2007.
7. Power, D. J., Supporting Decision-Makers: An Expanded Framework, In Harriger, A. (Editor), e-Proceedings Informing Science Conference, Krakow, Poland, June 19-22, 2001, 431-436 see http://proceedings.informingscience.org/IS2001Proceedings/pdf/PowerEBKSupp.pdf.
8. Power, D. J., Decision Support Systems: Concepts and Resources for Managers, Westport, CT: Greenwood/Quorum, 2002.
9. Arnott, D. and G. Pervan, "A critical analysis of decision support systems research", Journal of Information Technology, 20, 2, 2005, 67-87.
10. Matsatsinis and Siskos (2002), Intelligent support systems for marketing decisions, Kluwer Academic Publishers
11. Rajendra, Akerkar; Sajja, Priti (2009). Knowledge-Based Systems. Jones & Bartlett Learning.
12. Sowa, John F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations (1st ed.). Pacific Grove: Brooks / Cole.
13. The formal Structuring of Subject Domain for oil and gas industry IT applications / M.S._Chesanovskyy, V. I. Sheketa, V.M. Yurchyshyn, T.R. Styslo // XIII International Theoretical and Practiccal Conference «Modern problems of radio engineeting, telecommunications, and computer science (TCSET'2016)», (Lviv-Slavske, 23-26 february, 2016). – Lviv, 2016. – P. 503-505.
14. Modeling methodology for knowledge-based systems of wells drilling control / V. I. Sheketa, V. Y. Pikh, T. R. Styslo, M. S. Chesanovskyy // XIIIth International Conference «Perspective technologies and methods in MEMS design (MemsTech'2017)», (Lviv-Polyana, 20-23 th April 2017). – Lviv, 2017. – P. 56-58.
15. Tirri, H., Kontkanen, P., and Myllymäki, P. (1996). A bayesian framework for case-based reasoning. InProceedings of the Third European Workshop on Advances in Case-Based Reasoning, EWCBR, pages 413–427. Springer-Verlag.DOI: 10.1007/BFb0020627 62
16. B. C. Jeng and T. P. Liang, Fuzzy indexing and retrieval in case-based systems,Expert Systems with Applications, vol. 8, no. 1, pp. 135–142, 1995.
17. A. F. Rodrı´guez, S. Vadera, and L. E. Sucar, A probabilistic model for CBR, inProceedings of the Second International Conference on Case-Based Reasoning(ICCBR-97), Providence, RI, Springer-Verlag, Berlin, pp. 623–632, 1997.
18. Sheketa V. Formal methods for solving technological problems in the infocommunications routines of intelligent decisions making for drilling control // Sheketa V., Vovk R., Romanyshyn Y., Pikh V. , Pasyeka M.// in Proceedings of the 2019 IEEE International Scientific and Practical Conference Problems of Infocommunications Science and Technology.– PIC S&T′2019.– 08-11 October 2019.– Kyiv, Ukraine.– P.29–34.
19. Sheketa V.I. Predicate queries modification, as an tool to work with the knowledgebases of oil&gas subject domain / V.I. Sheketa // In proceedings of 6-th International scientific conference "Modern problems of radio engineering, telecommunications and computer science – TCSET'2004", (Lviv-Slavsko, February 24-28, 2004). – Lviv, 2004. – P.315–319.