Clarification KBS as Consultation-Justification Mash Ups
Proposing a Novel Paradigm for All-in-One Knowledge-based Systems

Martina Freiberg, Felix Herrmann and Frank Puppe
Department of Artificial Intelligence and Applied Informatics, Institute of Computer Science, University of Würzburg, Am Hubland, Würzburg, Germany

Keywords: Knowledge-based System, Clarification KBS, Consultation-Justification Mash Up, KBS UI Design, Agility.

Abstract: Regarding knowledge-based systems (KBS), the seminal paradigm—perfectly mimicking human experts—is gradually replaced by an increasing demand for enabling users to influence the reasoning process according to their domain knowledge. Therefore, we propose a novel KBS paradigm: *Clarification KBS* as a mash up type of consultation and justification interaction—intended to foster active user participation according to users' competency, the KBS' explicability, and the support for learnability. We introduce the theoretical concept of clarification KBS, as well as appropriate UI-/interaction variants. Further, we discuss the results of iteratively evolving and evaluating *ITree*, a specific clarification KBS implementation for the legal domain.

1 INTRODUCTION

Today, knowledge-based systems (KBS) are widely applied, both in research-based and industrial applications—e.g., for medical documentation/diagnosis, technical fault diagnosis, or environmental advice (Castellanos et al., 2011; Ting et al., 2011; Chen et al., 2012; Zeng et al., 2012). Yet, the once common paradigm, that a KBS should perfectly mimic the abilities of human experts, is gradually replaced—by the increasing demand for enabling users to influence the reasoning process according to their expertise, e.g., by answering expert shortcut questions. Another desirable objective for such KBS is to provide a high level of skill-building ability—i.e., to enable users to gain knowledge about the domain by using the KBS.

A typical KBS session commonly conforms to answering questions that are queried by the system—other KBS types, such as embedded KBS, also exist but are not targeted here. Based on the input data, the KBS typically derives one or more solutions; there, it is essential to also explain *how* and *why* the respective conclusion was drawn, i.e., providing a *justification*. Justifications basically support reproducing and reviewing the KBS's operation and conclusions (Puppe, 1993)—thus, apart from leveraging the KBS validation task, justifications help to build trust in the KBS, enable users to acquire domain knowledge, and thus enhance the overall user experience and utility. So far, handling the data input and presenting the results and corresponding justifications mostly are treated separately. In this paper, we motivate that mashing up the consultation and justification interaction can be highly beneficial: Resulting KBS offer a high level of explorability and freedom to the user; that in turn encourages users, to bring in their own competency and expertise; finally, the anytime provided justification, offered by such mashup KBS, strongly fosters the explicability and thus learnability of such a system. Today, diverse research is found regarding explanation generation and their use/utility (Arnold et al., 2006; Pinheiro et al., 2006); however, explanation/justification presentation, general KBS UI design and (usability) evaluation are still rather neglected. Specifically for clarification KBS, except our seminal work (Freiberg and Puppe, 2012) no similar research efforts are published so far. Therefore, as a generalization of our prior work, we propose *clarification KBS* as a consultation-justification mashup type in this paper.

The remaining paper is organized as follows: In Section 2, we propose the theoretical concept of clarification KBS as consultation justification mashups. Appropriate UI representations are presented in Section 3. We report various evaluations in Section 4, that we performed specifically for ITree—a particular clarification KBS instantiation. In Section 5, we conclude with a summary of the presented work and an outlook on prospective future research topics.

Freiberg M., Herrmann F. and Puppe F..
Clarification KBS as Consultation-Justification Mash Ups - Proposing a Novel Paradigm for All-in-One Knowledge-based Systems.
DOI: 10.5220/0005028501680175
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2014), pages 168-175
ISBN: 978-989-758-049-9

2 CLARIFICATION KBS TYPE

Forward Consultation & Backward Clarification. Forward- and backward chaining are known reasoning paradigms for rule-based systems, c.f., (Russel and Norvig, 2010, Ch. 6). Similarly, we define two KBS progression types: Starting with defined init questions, *forward consultation* KBS query in all directions, as all solutions contained in the KB can be derived equally well depending on the provided input. In contrast, *backward clarification* KBS target exclusively one defined solution at a time; i.e., only questions, that potentially contribute to the target solution rating, are posed.

Follow-up Questions & Abstractions. *Follow-up questions* are not contained in the original core interrogation sequence, but become activated only in case a certain trigger question is answered. *Abstraction questions* are questions the value/answer of which is automatically calculated by the KBS; an example is the *BMI*, which can be derived automatically once values for weight and size are known.

2.1 Clarification KBS—Base Concept

The core idea of clarification KBS is to unite consultation and justification interaction within an all-in-one UI module. They base on backward clarification, i.e., they target a single solution only. Clarification KBS generally can be seen as correspondent to the testing part in *hypothesize and test*: Hypothesizing—i.e., narrowing down the solution space to a probable solution candidate—can be performed using forward consulting KBS types or any other means such as mind-maps etc.; a clarification KBS then handles its detailed inspection.

Due to its mashup characteristic, the clarification KBS type can be invoked in two alternative ways: *Filled in*, and *empty*. Main objective of the *filled in* variant is to provide an interactively enhanced justification view with integrated, explorative consultation facilities. Thus, the target solution already possesses a certain rating/state, and the contributing question/answer pairs may be highlighted distinctly. Additionally, answer options might indicate—visually or textually—in what way they contribute to the target solution rating. The *empty* clarification variant aims at providing single issue consultation with integrated justification. Therefore, it is initialized with the yet unrated target solution, and entirely unanswered (and thus, not marked) questions. However they already might be marked with corresponding answer contribution indicators, see above. Those initialization variants can be applied for all UI styles that are proposed in the subsequent section.

3 CLARIFICATION KBS—THE UI

In the following, we introduce some interesting UI styles for clarification KBS: *Hierarchical Clarifier*, *Answer Form Add-On*, *Daily*, and *Interview* style. The latter two are equally well applicable for forward consultation; yet due to the focus of this paper, we describe them exclusively for the clarification context.

3.1 Hierarchical Clarifier Style

Hierarchical Clarifier is a single-line, interactive tree style, sketched in Fig. 1. It presents the question as well as all corresponding answers as one tree node, that spans (a greater part of) the UI. In the example, we experimented with placing answer options before the question text as to increase the efficiency of the answering interaction. The respectively clarified solution is displayed as topmost node (Fig. 1, a); this is followed by the base node set, representing all first-level items that directly contribute to the core issue rating (Fig. 1, b). Each question node again can

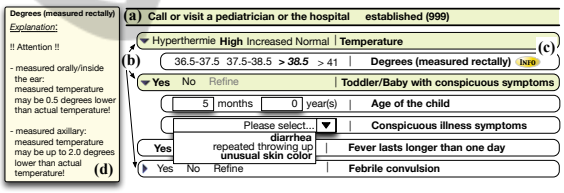

Figure 1: Hierarchical Clarifier Base Conception.

be followed recursively by further refinement levels; thereby, parents denote the *abstraction*, and their children the corresponding *abstraction sources* that derive the parent's value; e.g., in Fig. 1, the first question *Temperature* is an abstraction question, refined by a one-question abstraction source *Degrees?*. Answered and thus solution contributing questions are marked as to indicate the abstract contribution value, e.g., the two green top-level items in the example; the respectively selected answer options are displayed in bold print. Further, we recommend the inclusion of an anytime information display—here, realized by a side-panel (Fig. 1, d)—which presents additional explanations for a question once the corresponding info-button (Fig. 1, c) of a question is clicked. This form of presenting hierarchically nested/abstracted knowledge allows for answering questions either directly on the more abstract level, in case the users possess the required domain knowledge—which leads

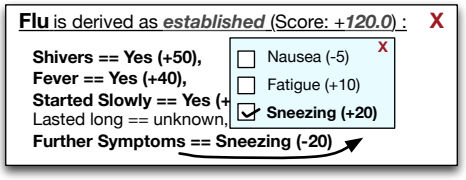

Figure 2: Answer Form Add-On enhanced Finding List.

to a shorter, more efficient session; or implicitly, by expanding the parent node, and answering the corresponding refinement questions—the respective answer values then are propagated to parent elements. That way, this style intends to support broad user groups (with varying proficiency) in highly expertise domains. It has to be noted, that technical or domain laymen may require some in-depth instructions and training time for familiarizing with this interaction principle. Hierarchical Clarifier optimally exploits its potential benefits in the context of highly expertise knowledge that can be defined using several differently detailed refinement levels; yet, this style can also be used with less staggered knowledge—then, however, it may degrade to a flat, list-like display that lacks any expert shortcuts.

3.2 Answer Form Add-on Style

Answer Form Add-on fore-mostly targets the context of providing interactive justification. Therefore, static justification presentation forms are enhanced by simple consultation widgets. In Section 3.2.1, we describe two exemplary justification presentation forms; based thereupon, we introduce the Answer Form Add-on style in Section 3.2.2.

3.2.1 Base Justification Styles

In general, KBS justifications can be presented using a vast array of text-based and visual techniques. Due to space restrictions, we introduce only two examples particularly apt regarding the Answer Form Add-On enhancement: *Finding List* and *Rule Graph*.

Finding List. We define a pair of a question and the user-entered answer as a *finding*. This justification style lists all findings that contribute to the target issue in a <question == value> style, c.f., Fig. 2; questions, that at the time of invocation are not (yet) answered, may display a default value, such as *unknown* (see Fig. 2, *Lasted long*). Already selected findings may further be highlighted, e.g., by bold face print; also, findings can indicate the value they contribute to the target issue—in the example, they are added in parentheses after the finding. Finding Lists generally suit

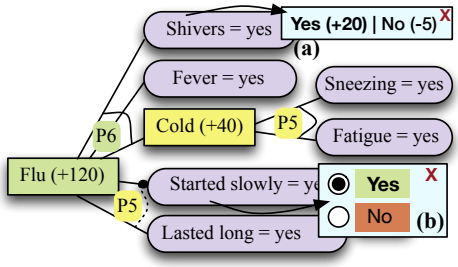

Figure 3: Answer Form Add-On variants for Rule Graph.

all knowledge types, that allow for the identification of questions and corresponding selected/non-selected answers—e.g., rule-based-, set covering knowledge, case bases, etc.

Rule Graph. Rule Graphs specifically visualize rule-based knowledge. Those are a generalization of *Derivation Graphs*, which have already been subject to prior research (Baumeister and Freiberg, 2010). Rule Graphs represent solutions as well as findings by distinct nodes, c.f., Fig. 3. In case a finding contributes to a solution rating, an arc is drawn between finding- and solution node. In the case of compound rules—that connect several findings by AND or OR conditions or that negate findings—the respective branches are connected by a rounded arc; in Fig. 3, a continuous line is used for the AND connection, a dotted line for OR connections, and a bold dot before the finding for a negation. Further, the rounded arc can be decorated with the rating, the rule assigns to the solution; e.g., the compound AND rule in the upper half of Fig. 3 rates the solution positively with the abstract rating class P6. When used as the basis for the Answer Form Add-on, it is crucial that a Rule Graph also visualizes those potentially contributing rules that have not yet fired in the session as to enable encompassing exploration. Such rules may be presented with a default answer value—e.g., *unknown*, or in a greyed manner for indicating their state.

3.2.2 Answer Form Add-on Conception

Answer Form Add-On enhances static justification presentation forms—see the previous section—by simple *Answer Form* popups. Therefore, presented findings serve as triggers that, once clicked, invoke a popup that displays all further answer alternatives for the question at hand. Fig. 2 sketches an example that is based on the Finding List justification. There, the question *Further Symptoms* offers the answers *Nausea*, *Fatigue*, and *Sneezing*; in the current session, Sneezing was selected, thus it is rendered in the Finding List—the further answers then are listed

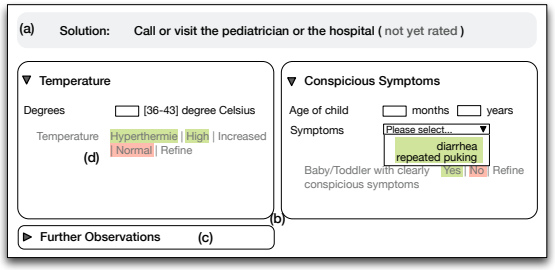

Figure 4: Daily Style Base Conception.

only in the answer form popup. Another example is shown in Fig. 3 for the Rule Graph justification style. It sketches two further variants for realizing the Answer Form popups themselves: In a plain juxtaposed style as shown in Fig. 3, a; and using radio buttons, Fig. 3, b; the latter variant additionally indicates the abstract rating value (green=positive, red=negative) of answers by background coloring. Once the user selects an answer inside the popup and closes it (X button), the respective finding is added in the current session and the justification view is re-rendered; this implies, that the original session state and justification are lost once the user starts interacting with the Answer Form widgets—it allows, however for a highly interactive, in-place adaption of the entered data. As Answer Form Add-On is based on justification presentation forms, it generally offers a high level of explicability and learnability: By explaining, why and how the current solution was reached; and by making explicit the coherences between findings and solutions.

3.3 Daily Style

Daily is a highly compact UI style where questionnaires, basically used as concept for grouping topically related questions, and their included questions form a column-wide, visual entity—similar to articles in newspapers—which becomes best evident, in multi-column variants; Fig. 4 sketches an example in two-column style. This style optimally supports the simultaneous display of (a greater part of) the entire KB. Questions and answers are rendered in a compact, juxtaposed style, c.f., Fig. 4, which in turn supports the overview and free exploration of all items. As the example indicates, the clarified solution is displayed prominently in the solution panel in the top or at one side of the UI (Fig. 4, a). Questions, potentially further grouped by collapsable questionnaires (Fig. 4, c), are rendered in the center part of the UI (Fig. 4, b). Answering questions is as simple as clicking on the respective answer text—repeated clicking deselects already set answers, denoting a highly intuitive overall interaction. Abstraction questions basically are handled inversely to Hierarchical Clarifier: Abstraction source(s) are presented as normal questions in the UI, whereas the abstraction value itself is displayed in an inactive (greyed) manner (Fig. 4, d)—once the user answers the abstraction source(s), the abstraction value is calculated, inserted, and displayed activated in case as users should be enabled to manually overwrite the derived value. In particular this implies, that there are no expert shortcuts as in Hierarchical Clarifier, but questions—in case they span several levels of abstraction—are always presented in their most fine-grained, detailed manner first. This on the on hand eases the KBS operation for rather inexperienced users (regarding the domain), on the other hand, there are no facilities for experts to render the KBS session more efficient. Thus, in general we assume Daily style more apt regarding inexperienced users. The example in Fig. 4 depicts an empty initialized clarification variant—i.e., the solution is presented yet without rating/state, and questions are only marked regarding their potential contribution to the solution, but without any answer selection-marks.

3.4 Interview Style

The core idea of the *Interview* UI style is to mimic human conversations. As a consequence, this style is only partly apt for clarification KBS: In the consultation-focussed context, i.e., the empty clarification variant, and in case users desire a high level of system guidance but do not attach as much importance on immediate, integrated justification. Interview basically adheres to a single question paradigm, i.e., it queries only one next question at a time, thus strongly mirroring one by one conversations. Thereby, the variant that the next question is presented automatically once the user answers the current question provides the most guidance and thus potential ease of use for laymen users; on the other hand, this could as well be perceived as a loss of control by other users. There, a stepwise navigation of already answered items, enabled by dedicated navigation buttons, denotes an alternative; i.e., the next suitable question is not displayed before the user actually clicks on a confirmation/navigation button (Fig. 5, c). Interview styles further offer the advantage of presenting extensive additional information in an automated, integrated manner (Fig. 5, a). Contextual information regarding the current progress of the interrogation can be provided either by a progress information display (Fig. 5, b); or by including an anytime solution display that presents the current state of the target solution (Fig. 5, d).

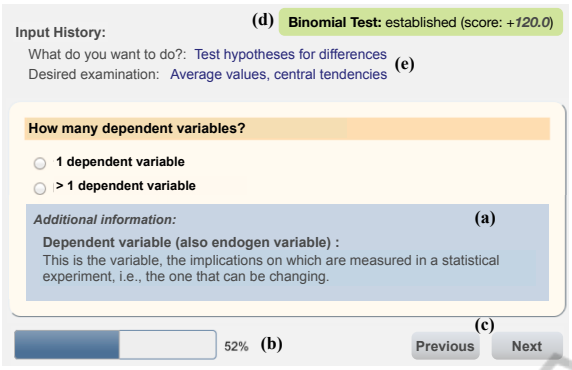

Figure 5: Interview Style Base Conception.

Despite its inherent focus on the consultation activity, Strict Interview can nevertheless be adapted for supporting a basic justification interaction: By integrating an interview object history that lists the entered findings (e.g., Fig. 5, e), and by enhancing that list by certain interactions supporting the adaption of such items. One example is interactive back-linking to the question display corresponding to the clicked finding, which enables users to quickly and easily go back and change the input; or the integration of answer form add-ons, as described previously, for in-place editing.

4 EVALUATION—ITREE

So far, we have implemented one instantiation of the Hierarchical Clarifier KBS style thoroughly—the *ITree* clarification KBS. Therefore, the tailored KBS engineering tool *ProKEt* (Freiberg et al., 2012) was used. The ITree vision originated from the JuriSearch project—a cooperation between the university of Würzburg and RenoStar, a german legal counseling company. Since then, the design and conceptualization of ITree have been further refined in several iterative development and evaluation cycles. In the following, we begin with a rough subsumption of the ITree conception and its first evaluations; for details, see (Freiberg and Puppe, 2012). We then focus on discussing the results of the latest evaluations and the consequential evolution of ITree.

Original ITree Vision, 1st & 2nd Evaluation. ITree is a tailored instantiation of Hierarchical Clarifier, see Section 3.1. It specializes that concept by mapping all questions on a fixed answer set—yes, no, uncertain; also, it applies special algorithms, based on partly complex AND/OR compound rules, for deriving the values of parent questions from the values of their child questions, see (Freiberg and Puppe, 2012).

The current implementation state of ITree is shown in Fig. 6. The highly proficient and complex legal domain is predestined for defining a deeply nested abstraction/refinement KB as used with Hierarchical Clarifiers; e.g., the ITree KB on dismissal contains up to 10 abstraction levels (three levels shown in Fig. 6). Therewith, legal ITree KBS support a wider range of users by offering manifold operation levels regarding the required prior domain knowledge.

The first evaluation in March 2012 showed, that iTree generally is a favorable clarification UI style, that basically supports an efficient and effective usage, free exploration, and skill-building ability. In comparison to an Interview variant, ITree overall was rated the more appropriate, convenient variant. However, problems occurred regarding the KB contents: Concerning the understandability of question texts, as too often specialist terms and intricate phrases were used without deeper explanation; but also concerning the refinement/nesting structures, see also (Freiberg and Puppe, 2012). As those two aspects had been the main (negative) cause variables, we experimented with two follow up variants of the initial KB: One especially targeted at legal experts (EXP—structured similarly to the original yet refined regarding the wording/explanations); and one targeted towards laymen (USER—restructured entirely as to provide items, most interesting to laymen, in a more prominent manner). Those two variants were evaluated comparatively in May 2012—therefore, legal laymen and expert users were asked to solve some legal cases with the KBS variants. Interestingly, the preference regarding the USER variant by laymen was not as evident as assumed: In total, the USER variant was preferred in 35% of the cases; the LEGAL variant in about 30%; most remarkably, however, also in 35% a complete indifference between both variants was stated, both by expert and laymen users. As the EXP variant is based on commonly taught, unified, legal schemes, and thus can be formalized (by the experts themselves) and maintained in an easier manner, we finally decided to further rework and refine exclusively the EXP variant, as there were no convincing indications in favor of the USER variant. As a basically positive result, the overall success rate regarding the evaluated cases increased to 61% (USER) and 72%(LEGAL), which denotes a good improvement from the success rate of 43% for the ITree style in the first evaluation.

3rd User Study and Consequential UI Adaption. After further refining the legal (expert oriented) KB as well as the basic ITree design, another user study was conducted at the end of 2013 with 19 students of med-

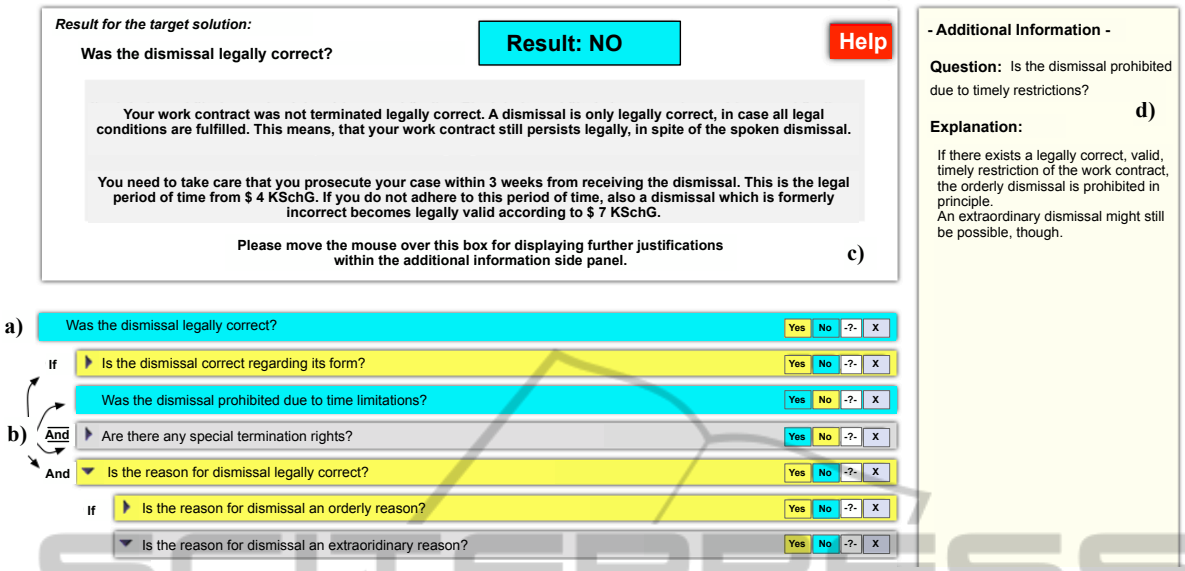

Figure 6: ITree Style Base Conception, used with a clarification KB on the legal correctness of a dismissal.

ical informatics; Similar to the prior studies they were asked to solve given legal cases with the KBS. Most remarkably, now a success rate of 90-100% could be achieved—a huge improvement compared to former evaluations. Also further aspects, rated on a scale from 1/very good–6/very bad received good results—e.g., the ease of use (2.0–2.5), the efficiency (2.2–2.5), overall ITree utility (2.1) or the KB contents (1.8–2.6), varying between the different cases to be solved. However, also this study revealed some specifically UI/implementation specific issues. The major remark concerned the (too unresponsive) loading speed of the system, both initially and when users enter data; there we reworked the entire mechanism as to significantly increase the loading behavior. Also, the side panel was (in some cases) perceived as too small—and thus enlarged in the ITree implementation.

Expert Usability Evaluation (4th Study). In early spring 2014, ITree finally also was assessed particularly focussing on its general usability and design (and not so much on success rates). Therefore, an expert evaluation study was performed in early spring 2014, where ITree (amongst further KBS UI styles) was evaluated. 30 evaluators—human computer interaction students of the university of Würzburg—performed both a heuristic evaluation (Nielsen, 1994) and a cognitive walkthrough (Wharton et al., 1994). This evaluation partly confirmed previous evaluation insights, partly exhibited interesting new findings.

The first remark concerned the status mediation by color coding: Adhering to a red/green/yellow traffic lights scheme, the solution and question state were highlighted according to the current solution derivation state or the question's contribution to the solution rating, respectively. Depending on the respective question, the answer yes and no can contribute both to a positive, or to a negative solution rating—which accordingly sometimes leads to coloring *yes* with green (positive contribution) or *yes* with red (negative contribution). The main remark there was, that this unsteady coloring of answer options was unfavorable, all the more in those cases where *yes* was marked red (as this additionally contradicts the cultural semantic feeling of yes being positive/green). Despite highlighting in the instructions that colors do *not* mirror the answer semantic but represent the contribution of that answer to the solution, this was rated as highly confusing. As color coding specifically was intended to mirror the current system status clearly at first glance—which was also approved of by several evaluation remarks—we did not remove colors entirely; also, we assumed that adding color-independent signs such as + and - would not be quite as clear and thus increase the memory load of users when estimating the KBS status. Thus, we introduced a different coloring scheme yellow/white/blue, shown in Fig. 6.

The next aspect was the scrolling behavior: Basically, ITrees are scrollable as to enable users to reach all nodes even if the expanded tree grows larger than the available UI space; however, in the original version this both induced a disappearance of the solution display header and of the additional information display side panel (which is especially critical as without the add-on explanations displayed there, ITree becomes almost unusable for most users). Thus, we fix-

ated and enlarged the side panel for prominent, anytime explanation display. Regarding the top panel, we added a flexible jump-to-the-top mechanism: Once users have answered enough questions for deriving a solution rating, the display automatically scrolls up to the solution panel; thus, the tree can to take up more available UI space as long as not solution is derived, yet as soon as a rating is available, it is brought prominently back into mind by the jump-scroll mechanism.

Regarding that side panel, also its general interaction—automatically updating panel contents, i.e., additional information for questions, when hovering a question node—was criticized. Specifically regarding high-resolution displays, the required precise hovering of potentially large question nodes was perceived as cumbersome and disrupting. However, we discarded the idea of using dedicated icon/question clicks as triggers, as ITree is deliberately intended as highly efficient UI and interaction type; thus requiring an extra click each time such information is required—potentially often in the complex target domains—would reduce the efficiency drastically. Thus for enhancing the side panel interaction, a short delay was added—i.e., once information for a question is displayed, hovering/touching another question does not update the panel immediately but only after a certain delay; this obviates unwanted explanation updates in the major part of cases.

The next remark concerned the initial automated expansion of parts of the tree—basically intended to highlight questions that are, on average, most relevant for a solution and thus should not be overseen. The evaluators remarked, however, that users often might be left wondering whether this a presentation flaw or a deliberate design decision. However, as we still believe that this initialization is beneficial especially for domain-related, experienced users—who might check such a similar base answer set in most cases anyhow—only a clear remark explaining the issue is added in the instructions but the mechanism was kept. This remark, as well as a short subsumption of the usage paradigm and color semantics are further displayed prominently in the UI header at system start—as initially not yet much solution related information is available for display there, this offers a great opportunity for presenting such important, initial information for quick start support.

Another major remark concerned the knowledge connector items *If*, *AND*, and *OR*, displayed in front of the questions c.f., Fig. 6, c for visually mirroring the underlying derivation knowledge (rules). Those were partly rated as confusing and as rather increasing the mental workload of users. For ITree as a clarification KBS, however, this in-place justifications are essential for increasing its on-the-fly explainability. Therefore, a short-hand key to such symbols is planned for minimizing potential confusion.

It was further criticized that invoking a *new case [Neuer Fall]* by clicking the respectively termed button does not display any warning regarding the consequential loss of all currently entered data. Basically, we did not assume that as critical due to the button label that already indicates that implicitly. As this was a generally rather frequently stated issue, we nevertheless added a confirmation popup which highlights this issue more prominently.

Recent Studies Summer 2014 and Synopsis. In summer 2014, two more large- and smaller scale ITree studies were performed (248 and 21 computer science student participants, respectively), thus following a highly agile, iterative development–evaluation cycle. The overall baseline ratings remained positive also in those last studies; e.g., overall utility (2.2–2.7), KB quality (2.6–2.8), knowledge mediation (2.6–2.7) or ease of use (2.5–3.0) on scales from 1/very good–6/very bad. Success rates quite constantly remained at 86–88%, which we account as great results given the highly expertise, specific domain and complexity of potential cases. Those studies further confirmed the suitability of the unusual coloring scheme and the added value of adding the delay regarding the additional information panel update. The latest suggestions for improvement, partly already realized, included: Moving the side panel as well as the answer buttons to the right side as to enable the western cultural left-to-right read flow; prominently displaying a notification once a solution rating is reached, in addition to the jump-scroll mechanism; and providing notifications once a user tries to enter values that contradict already derived values.

Having evolved iteratively over about two years, ITree now exhibits rather good baseline ratings; this specifically concerns the success rate, but also further important traits such as utility, KB quality, knowledge mediation, ease of use, etc. Also, the overall design/interaction was mostly perceived as convenient. This makes us confident, that ITree meanwhile denotes a beneficial clarification KBS UI style; and consequently, that the assumed benefits of consultation/justification mashup types—such as high flexibility, bringing in own competency, skill-building ability—within a unified KBS solution actually apply.

As one base insight, we claim that especially in the context of highly expertise KBS, an iterative, agile development paradigm is extremely beneficial—which

was strongly fostered by the tool *ProKEt*. Another important general insight was, that the evaluation results not only depend on the particular UI/interaction realization, but likewise also on the KB quality *and* the quality of the test cases/descriptions—at least in similarly expertise domains and consequential complex test case descriptions.

5 CONCLUSIONS

In this paper, we proposed *Clarification KBS* as a novel KBS paradigm. Clarification KBS not only mashup consultation and justification UI/interaction within a single UI; they particularly possess the potential to foster more active user participation by letting them bring in their own competency and further offer a high explicability—therewith providing domain- and KBS related skill-building ability. We first introduced the basic concept of clarification KBS. We further proposed some interesting, suitable UI representations: *Hierarchical Clarifier*, *Answer Form Add-On*, *Daily*, and *Interview* style. Also, we discussed the results of evolving and evaluating an *ITree* KBS UI—a tailored instantiation of Hierarchical Clarifier—in the legal domain.

The overall results showed that ITree in particular and thus clarification KBS in general can stand up to the assumptions regarding the assumed benefits consultation/clarification KBS mashups; yet, also the insight manifested that clarification KBS are not well applicable for all diverse user types likewise—rather, they seem particularly useful for users with at least a little domain-related and/or technical experience; this assumption will be subject to future work. Further practical work regards the implementation of a more general clarification KBS for the medical domain. The currently envisioned medical clarifier requires the flexible integration of diverse further question types—e.g., numerical, date, or free-text questions. Consequently, also the propagation algorithm of ITree will require some reconsideration and generalization.

Also, the thorough implementation and evaluation of the other proposed clarification UI types is subject to future work. Regarding the Answer Form Add-On style, this particularly also concerns the static base justification types, two of which were presented in this work; there, we are currently realizing some experimental solutions with the tool ProKEt. Finally, also the envisioning and realization of further novel clarification KBS UI styles offers ample room for future work.

REFERENCES

Arnold, V., Clark, N., Collier, P. A., Leech, S. A., and Sutton, S. G. (2006). The differential use and effect of knowledgebased system explanations in novice and expert judgment decisions. *MIS Quarterly*, 30(1):79–97.

Baumeister, J. and Freiberg, M. (2010). Knowledge visualization for evaluation tasks. *Knowledge and Information Systems*, 29(2):349–378.

Castellanos, V., Albiter, A., Hernández, P., and Barrera, G. (2011). Failure analysis expert system for onshore pipelines. part-ii: End-user interface and algorithm. *Expert Systems with Applications*, 38(9):11091–11104.

Chen, Y., Hsu, C.-Y., Liu, L., and Yang, S. (2012). Constructing a nutrition diagnosis expert system. *Expert Systems with Applications*, 39(2):2132–2156.

Freiberg, M. and Puppe, F. (2012). iTree: Skill-building User-centered Clarification Consultation Interfaces. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2012)*. SciTePress Digital Library.

Freiberg, M., Striffler, A., and Puppe, F. (2012). Extensible prototyping for pragmatic engineering of knowledge-based systems. *Expert Systems with Applications*, 39(11):10177–10190.

Nielsen, J. (1994). Heuristic Evaluation. In Nielsen, J. and Mack, R. L., editors, *Usability Inspection Methods*, pages 25–62. John Wiley & Sons, New York.

Pinheiro, V., Furtado, V., Silva, P. P. D., and Mcguinness, D. L. (2006). Webexplain: A upml extension to support the development of explanations on the web for knowledge-based systems. In *Proceedings of the Software Engineering and Knowledge Engineering Conference, San Francisco*.

Puppe, F. (1993). *Systematic Introduction to Expert Systems*. Springer-Verlag. ISBN: 3-540-56255-9.

Russel, S. J. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Pearson. ISBN 13: 978-0136042594.

Ting, S., Kwok, S., Tsang, A. H., and Lee, W. (2011). A hybrid knowledge-based approach to supporting the medical prescription for general practitioners: Real case in a hong kong medical center. *Knowledge-Based Systems*, 24(3):444–456.

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). Usability inspection methods. chapter The Cognitive Walkthrough Method: A Practitioner's Guide, pages 105–140. John Wiley & Sons, Inc., New York, NY, USA.

Zeng, Y., Cai, Y., Jia, P., and Jee, H. (2012). Development of a web-based decision support system for supporting integrated water resources management in daegu city, south korea. *Expert Systems with Applications*, 39(11):10091–10102.