Pattern-driven Knowledge Systems Engineering

Martina Freiberg and Frank Puppe

Department of Artificial Intelligence and Applied Informatics, Institute of Computer Science, University of Würzburg, Am Hubland, D-97074 Würzburg, Germany
freiberg/puppe@informatik.uni-wuerzburg.de

Abstract. Despite increasing relevance in research- and industrial contexts, the implementation of knowledge-based systems (KBS) still is a challenging task. We motivate, that patterns—basically a renowned means for providing reusable solutions for similar problems—can drastically leverage development efforts and time. In this paper, we propose a framework for pattern-driven, encompassing KBS development, consisting of: Tailored usability criteria for a clear delimitation of KBS solutions, a basic pattern specification template, and a collection of foundational KBS UI patterns. We further describe practical experiences with the approach, entailing: The reference implementation of several patterns with the tailored development tool ProKEt, their design- and usability-related evaluation, and empirical evidence of applying pattern-driven KBS development in actual projects.

1 Introduction

Despite increasing relevance in research- and industrial contexts, knowledge-based systems (KBS) engineering still denotes a challenging task. In general software engineering, patterns are renowned for describing proven solutions and preventing common pitfalls, thus fostering reuse and strongly leveraging overall development efforts. To date, various pattern collections for general UI- and interaction design are proposed, including [7, 8, 9]; also, many resources are available freely on the web.

In contrast to standard web pages or query forms, KBS do not solely build on strictly predefined question sequences; rather, they characteristically live from follow-up items—flexibly included interview items that become relevant only during the questioning session and based on the concrete user input. Also, KBS often require the prominent integration of additional information for elaborating the knowledge base / interview items more clearly and of (in-place) explanations of the results. This calls for tailored UI and interaction solutions that best support those requirements.

We motivate, that also KBS engineering can strongly profit from fitted patterns that subsume such proven (and optimally evaluated) KBS solutions; this can strongly support and refine requirements engineering, and leverage encompassing KBS development. First steps into that direction have been already taken with regards to the knowledge base, e.g., [6]. As counterpart, we propose tailored

patterns, that capture foundational KBS interaction and UI design solutions regarding various contexts and target objectives; to the best of our knowledge, no similar efforts have been published so far.

The rest of the paper is organized as follows: In Section 2, we introduce a basic KBS pattern specification framework: A short definition of relevant terms, tailored (usability-related) classification criteria, and a KBS UI pattern specification template. A collection of four framing patterns, that can be fine-tuned into a total of ten pattern variants, is proposed in Section 3. Practical experiences related to pattern-driven KBS development are reported in Section 4, and a summarizing conclusion and promising future work are presented in Section 5.

2 KBS UI Pattern Specification Framework

Before proposing a set of usability-related KBS delimitation criteria in Section 2.1 and sketching a basic pattern specification template in Section 2.2, we clarify some basic terms used in the remaining paper.

Forward- & backward progression: *Forward progression* starts with an empty solution set; from one or more init questions, such a KBS then questions in all directions, depending on the particularly implemented indication mechanisms. In contrast, a *backward progression* KBS is initialized with a target solution and poses only those questions that contribute to the final state of that chosen solution.

Multiplex consultation- & clarification KBS: *Multiplex consultation KBS* use forward progression, whereas *clarification KBS* base on backward progression. Clarification KBS can further be used with two application focuses: Consultation focussed—i.e., the clarification KBS is started empty, and all contributing questions are posed. Or justification focussed—then, such a system is called for justifying a solution that already has been derived in the preceding session, thus corresponding to an elaborate, interactive solution explanation.

2.1 Tailored, Usability-related KBS Classification Criteria

Today, diverse UI design- and usability guidelines and standards are available, such as Nielsen's heuristics [5] or the universal design guidelines of Lidwell [4]. However, those typically are defined rather generally as to be applicable for diverse interactive software system types. Due to their specifics as mentioned in the introduction, KBS require more tailored criteria; those then can be used for clearly delimitating their specific characteristics—as, e.g., for the pattern specification in this work—or for rating KBS solutions regarding their quality and usability. We propose eight tailored, usability-related classification criteria as follows:

1. Compactness: *How many interview items are presented simultaneously?*

2. Comprehensibility: *Is support provided for understanding specialist, complex, or ambiguous knowledge base contents (additional explanations, surrounding, contextual questions), and in learning something about the domain?*

3. Descriptiveness: *Does the KBS suggest how respective questions/answers influence the final result of the session, e.g., by indicating the score (change)?*

4. Efficiency: *How long does a characteristic session take and how many interview items need to be processed?*

5. Explorability (Participation): *Are users enabled to deviate from the suggested questioning sequence, are potential expert shortcuts provided?*

6. Intuition (usage): *Are the applied presentation/interaction forms familiar or otherwise self-descriptive? If not, are particularly novice users supported (instructions, tutorials, examples)?*

7. Transparency: *Is the current state (i.e., state of questions, results, overall progress) clearly and comprehensibly mediated at any time?*

8. Clear Arrangement/Aesthetics: *Does the overall design exhibit certain aesthetics, e.g., by using a small number of virtual lines & basic symmetry?*

2.2 KBS UI Pattern Specification Template

Table 1 summarizes a basic template for specifying KBS UI patterns in a unified and clear manner. All variations of a base pattern exhibit some similar characteristics, e.g., the core KBS objective. They vary regarding the specific realization of the UI/interaction, the extent of adhering to KBS classification criteria (see Section 2.1), the target users, knowledge specifics, and the imposed consequences. In the following pattern descriptions, we only provide a summarizing template item *variations* that subsumes specifics regarding the UI/interaction and required knowledge; we further delimitate the differences regarding the classification criteria and the target users in Table 2; consequences, as well as details on the example implementations, are omitted here due to space restrictions.

Pattern Section	Description
Problem Statement	Specifies the problem, that is solved by this pattern, based on the the tailored KBS usability criteria as described in Section 2.1.
Solution	Describes the general (UI and interaction) solution that all variants of this pattern apply, e.g., the characteristic base interaction.
Variations	Variations of the fundamental pattern, differing regarding: The targeted *user types*, the specific *UI realization*, *knowledge specifics*, (*consequences*, and *example implementation details*—not elaborated in this paper).

Table 1: Basic template for specifying fundamental KBS UI patterns.

3 KBS UI Pattern Collection

We propose four basic KBS UI patterns: *Questionnaire*, *Interview*, *Clarifier*, and *Clarifier Hybrid*, along with variants. In earlier research, we already introduced

three basic interaction categories for KBS, see [1]; there, *Adaptive Entry* maps to *Questionnaire*, *Guided Entry* to *Interview*, and *Autonomous Entry* to *Clarifier*. The patterns basically are intended independent from a specific knowledge representation—in the sense that for the pattern/UI it is irrelevant whether a rule-base or a covering model derives the solution ratings; however, some patterns are favorable regarding specific knowledge characteristics—e.g., CheckList Questionnaire requires all questions to be mappable on a fixed answer set; we included short remarks on such specifics in the variations' descriptions.

3.1 Questionnaire Pattern

Problem Statement: The KBS should *compactly* display a greater part of the KB, offer a *highly transparent UI*, *intuitive usage*, and a c*ertain extent of explorability*; comprehensibility, is no key requirement for the core UI itself.

Solution: Questionnaire KBS resemble paper- or web-based questionnaire forms. Depending on the particular UI style, many to all indicated interview objects are displayed simultaneously and typically ordered in some form of grid-based layout. Questionnaire may suggest (visually), but does not necessarily prescribe, an optimal interrogation sequence and thus fosters explorative usage. A certain comprehensibility can be achieved by adding auxiliaries—such as informative popups with additional explanations for interview items. Per default, Questionnaire uses forward progression, c.f., Section 2

Variations: Box-, Daily-, and CheckList Questionnaire.

a. Box Questionnaire closely adheres to the design of standard questionnaires by using established, familiar question presentation forms—e.g., checkboxes and radio buttons for choice questions, see Figure 1, I; thereby, each question is rendered within a distinct box, resulting in a very regular layout, but some waste of UI space.

b. Daily Questionnaire, originally inspired by daily newspapers, realizes a more flat, juxtaposed presentation style for questions and answers, c.f., Figure 1, II; therefore, each question along with its answer options is placed in a line, implying a less regular yet more compact layout than the Box variant.

c. CheckList Questionnaire mirrors paper-based check lists by representing answer options by columns that simply are ticked off, see Figure 1, III. Therefore, all choice questions need to be mappable to a fixed answer set; including further types, e.g., numerical questions, is possible, yet best results regarding efficient interaction and compactness are achieved with choice questions only.

3.2 Interview Base Pattern

Problem Statement: The KBS UI should be *highly intuitive* and *easily comprehensible*, thus specifically supporting novice users/domain laymen; in turn, compactness, descriptiveness, efficiency, explorability, as well as UI transparency can be neglected.

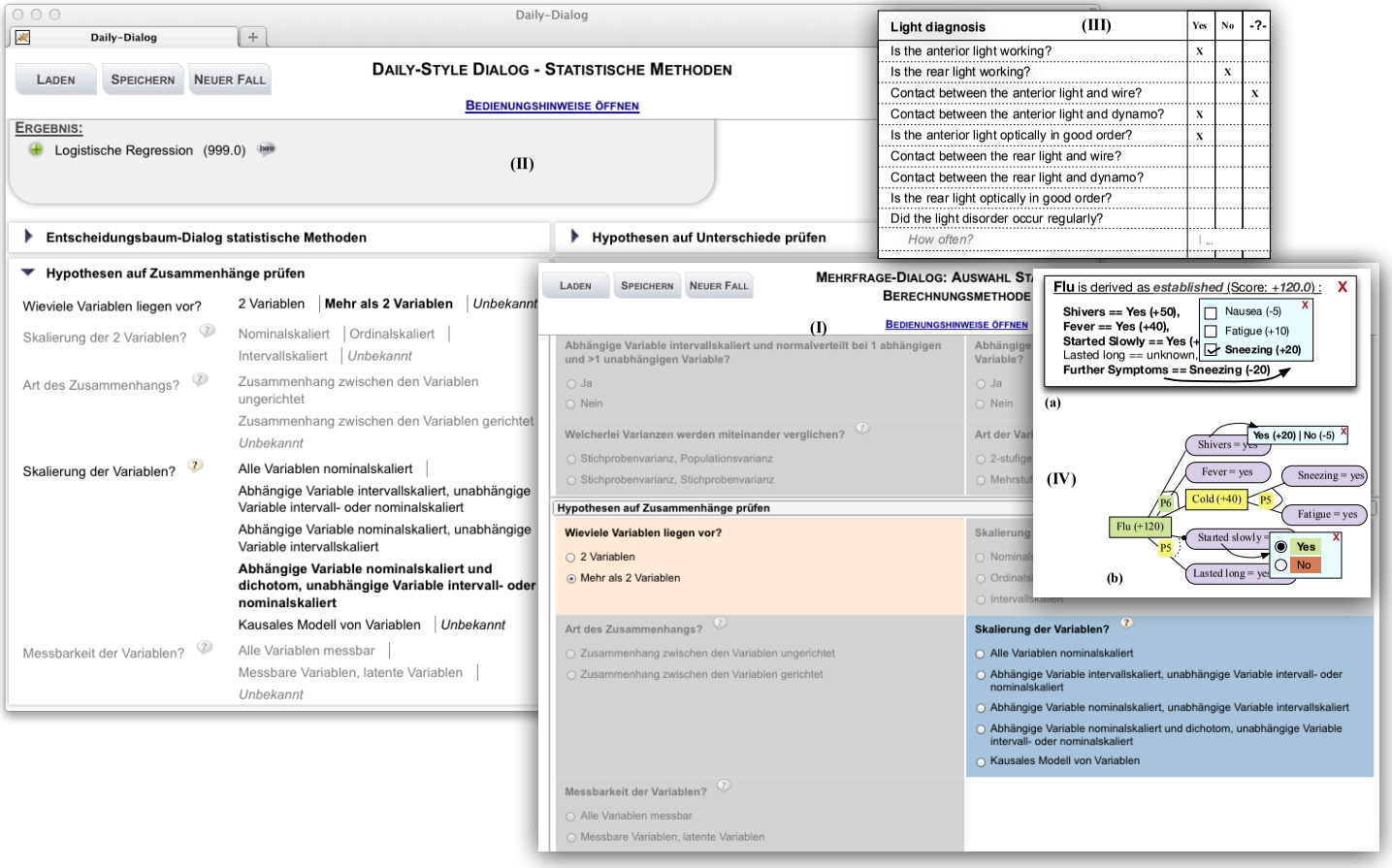

Fig. 1: KBS Patterns—examples 1: Box Questionnaire (I) and Daily Questionnaire (II) as implemented in ProKEt, CheckList Questionnaire (III), and Form Add-on Clarifier (IV) as preliminary sketch.

Solution: Interview imitates human conversations by presenting always a single question—or a group of related questions—at a time; additional information are available anytime quickly and easily, e.g., by directly integrating it near to the corresponding interview object. Interview typically prescribes the interrogation sequence in a rather fixed manner. The basic lack of transparency can be alleviated by integrating auxiliaries such as an interview object history—listing the already processed interview items—and a progress information display.

Variations: Strict-, Grouped-, and Hierarchical Interview.

d. Strict Interview displays only a single question at a time together with its additional information, see Figure 2, I. Thus, optimally, the KB should provide suitable auxiliary information for each interview item. Further, a sophisticated indication mechanism is advisable for keeping the possible interrogation paths at solid lengths, especially regarding large KBs.

e. Grouped Interview sequentially displays groups of (optimally topically related) questions or single questions; thus, it is a bit more efficient than Strict Interview, and offers more explorability, as the particular sequence within question groups is not prescribed. The UI uses a similar basic frame as Strict Interview, where Questionnaire variants are used for rendering the groups, c.f., Figure 2, II.

f. Hierarchical Interview offers an interactively navigable tree UI specifically for decision tree knowledge, c.f., [6]. Answer options are presented as width-spanning tree nodes, c.f., Figure 2, III. Clicking on answer nodes induces their expansion by the answer options of the follow up question. Solutions are represented by distinct nodes at the final nesting levels; thus, the tree path from outer nodes to the solution particularly enables users to 'read' the justification of a solution from the visual tree structure. Auxiliary information is presented in a dedicated side panel—either on click or by hovering the nodes.

3.3 Clarifier Base Pattern

Problem Statement: Selected single solutions in highly expertise domains should be investigated exclusively and in-depth. The KBS UI should be *compact*, *transparent*, *descriptive*, and offer skill-building ability induced by a *high comprehensibility* of the contents; users should be enabled to increase efficiency in contributing their personal knowledge, e.g., for using shortcuts regarding the interrogation sequence (*explorability/participation*). Intuitive usage, in contrast, is no key requirement.

Solution: Clarifier characteristically uses backward knowledge, see Section 2, for investigating only a single issue at a time. Therefore, Clarifier renders the target solution and all contributing questions—i.e., that potentially alter the solution rating in any way—simultaneously and offers means to adapt answers and thus investigate the consequences on the solution.

Variations: Hierarchical Clarifier and Form Add-on Clarifier.

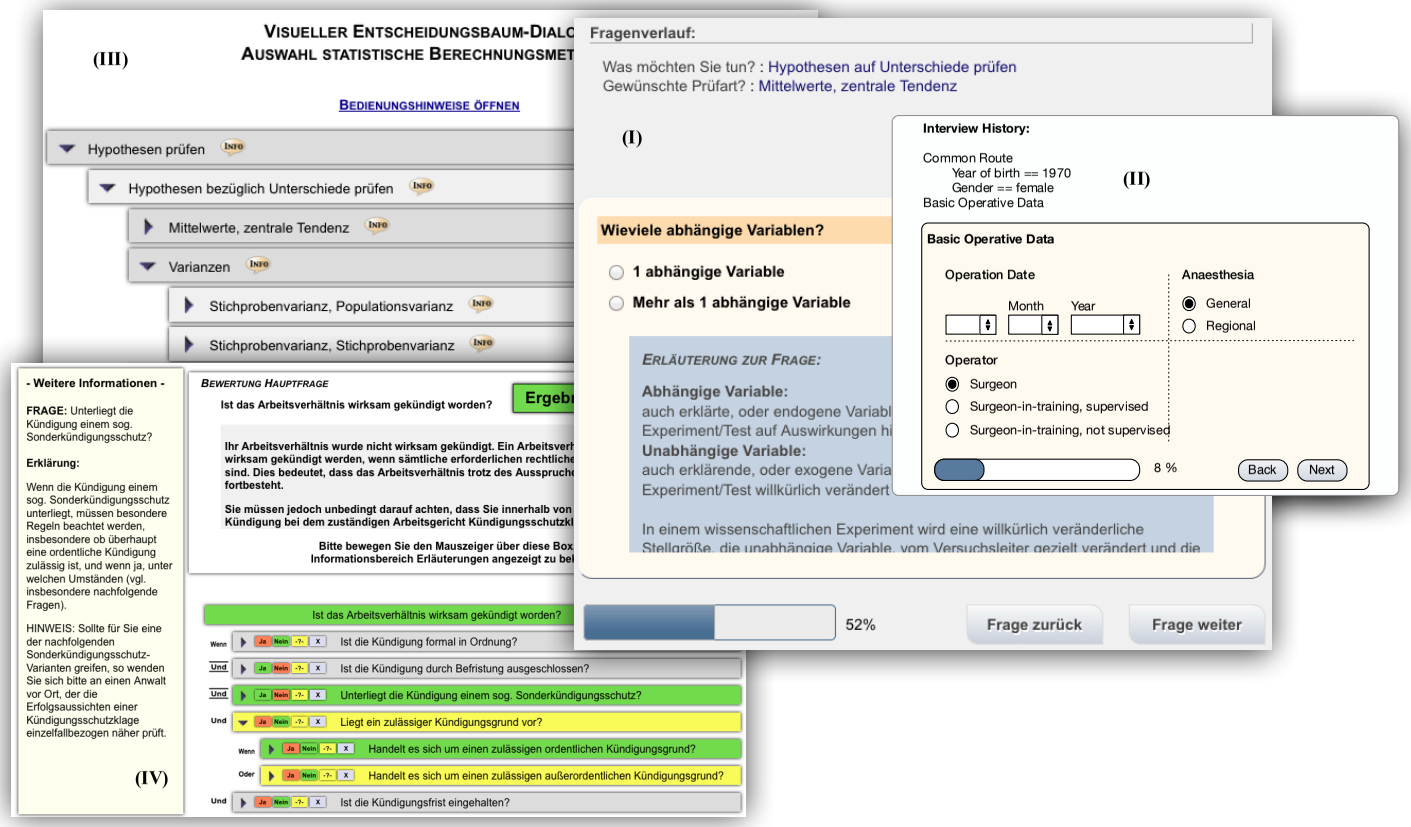

Fig. 2: KBS Patterns—Examples 2: Strict Interview (I), Hierarchical Interview (III), and Hierarchical Clarifier (IV) as implemented in ProKEt, Grouped Interview (II) as preliminary sketch.

g. Hierarchical Clarifier exhibits its strengths optimally with backward knowledge that is refined over several abstraction levels. It displays question and answers options within one node of its tree-style UI, see Figure 2, IV. The topmost node corresponds to the target solution, and is followed by nodes that correspond to the directly solution-relevant questions. Each question node again can be followed recursively by further question levels where the children denote a more fine-granular partition—one or several questions—of its parent. Thus, users decide whether to answer the more abstract top-level questions; or whether to implicitly answer them by expanding them and processing the children—child answers then are propagated recursively back to the parents.

h. Form Add-on Clarifier adds minimalistic consultation widgets to static base justification presentation forms, such as finding lists or rule graphs, c.f., Figure 1, IV a&b. Clicking the interview objects in the justification automatically triggers compact (popup) consultation widgets; those contain all answer options for the respective question, potentially also indicating the value that is added to the solution. This allows for interactively adapting answers and thus exploring and clarifying a selected solution based on its general justification view.

3.4 Clarifier Hybrid Pattern

Problem Description: A more intuitively usable and easily comprehensible UI representation for using clarification knowledge is desired.

Solution: Clarifier Hybrids merge intuitive, comprehensible KBS UIs with backward knowledge for supporting especially also novice or laymen users in using clarification KBS. Both Questionnaire and Interview patterns are suitable for using backward knowledge. The base implementation of Clarifier Hybrid then corresponds to the variants described in Sections 3.1 and 3.2; in contrast to that, the targeted backward knowledge is processed, which might—depending on the actually selected UI variant—require some additions to the base UI; for example, widgets for explicitly navigating the hierarchical refinement levels.

Variations: Clarification Interview and Clarification Questionnaire.

3.5 Pattern Variants—Detailed Delimitation

Table 2 summarizes the fine-grained delimitation of the proposed patterns according to the tailored classification criteria, introduced in Section 2.1; the extent of their realization is rated from low (L) to high (H). Further, corresponding target user characteristics for each pattern are specified by classifying the domain-expertise and the frequency of usage. Thus, the delimitation table serves as quick reference which pattern to apply in what context. If, e.g., a KBS solution is requested that is both highly intuitive usable, and—given the appropriate knowledge—highly comprehensible, also and especially for first-time users, then the Strict Interview pattern suits best, c.f., Table 2.

		Questionnaire			Interview			Clarifier		Hybrid	
		3.1.a	3.1.b	3.1.c	3.2.d	3.2.e	3.2.f	3.3.g	3.3.h	3.4	3.4
Usability Criteria	**compact**	M	H	H	L	M	H	H	H	L	M
	comprehensible	M	M	M	H	H	L-M	M-H	M	M-H	M-H
	descriptive	L	L	L	L	L	L	H	H	L	L
	efficient	M	H	H	L	M	M	M	L-M	M	L-M
	explorable	H	H	H	L	L	M-H	H	H	L	H
	intuitive	H	M	H	H	M	L-M	L	M	M-H	H
	transparent	H	H	H	M	M	M	H	H	L-M	H
	clear arranged	H	M	H	L	M	L-M	M	L-M	L	M
Users	**expertise: laymen**	X	X	X	X	X	(X)	/	(X)	X	X
	experienced	X	X	X	X	X	X	X	X	X	X
	expert	(X)	X	X	/	(X)	X	X	X	(X)	X
	usage: one-time	X	X	X	X	X	(X)		(X)	X	X
	frequent	X	X	X	(X)	X	X	X	X	(X)	X

Table 2: KBS UI pattern delimitation according to tailored usability-related KBS classification criteria with extent of their realization—High (H), medium (M), and low (L)—and further, regarding two user characteristics—expertise, and usage frequency.

4 Practical, Pattern-Driven KBS Development

Our practical experiences related to KBS UI patterns encompass: Support of pattern-driven KBS development by the tailored tool ProKEt [3]; a usability assessment of selected pattern reference implementations with ProKEt; and several current projects where KBS patterns/pattern-driven development was beneficial.

4.1 ProKEt: Tool Support for Pattern-driven KBS development

We already introduced ProKEt as a tailored KBS engineering tool that specifically supports template-based development in [3]: By realizing the KBS UI framework through defining highly modular (HTML) templates with varying complexity, that can recursively be assembled into more comprehensive ones. That main mechanism still persists, yet the collection of supported templates and readily available KBS patterns has been extended. Currently supported are Box- and Daily Questionnaire, along with encompassing options for simple property-based fine-tuning—e.g., regarding whether to hide non-indicated items; Strict- and Hierarchical Interview, with an optional interview history and progress bar; a configurable solution panel display for all those variants; and Hierarchical Clarifier, with a tailored add-on information display and a specific solution panel variant.

4.2 Evaluation of Selected KBS UI Patterns

In early 2014, an encompassing evaluation was conducted regarding ProKEt reference implementations of the following KBS UI patterns: Box-/Daily Questionnaire, Strict-/Hierarchical Interview, and Hierarchical Clarifier. Therefore, first

an expert evaluation was conducted by 30 HCI students, using heuristic evaluation according to Nielsen [5] and the cognitive walkthrough technique [10]; the basic goal was to assess the demo implementations regarding their basic quality and usability. Afterwards, in total 248 computer science students participated in a more comparative user study, where certain given problem descriptions were to be solved with each of the tested KBS; there, students were further instructed to fill in a short questionnaire for collecting some basic unified values—e.g., regarding the overall utility of the KBS—and to provide informal feedback.

Evaluation Item	**Daily**	**Box**	**HInterv**	**SInterv**	**HClari**
Success Rate = Correctly solved/all cases	**88.71**	**91.53**	**27.02**	**20.16**	**88.31**
Questionnaire Items					
Q1. Overall utility of the system	**2.04**	**1.93**	**3.63**	**3.06**	**2.72**
Q3. Belief in the result's correctness	**1.68**	**1.76**	**4.13**	**3.86**	**3.03**
Q4. KB quality=content,structure	**2.24**	**2.16**	**3.73**	**3.08**	**2.82**
Q5. Knowledge mediation	**3.79**	**3.77**	**3.64**	**3.18**	**2.78**
Q6. Perceived ease of use	**1.95**	**1.57**	**3.30**	**2.28**	**3.03**
Q7. Efficiency of the KBS	**2.01**	**1.84**	**3.45**	**2.86**	**2.83**
Q9. Rating of the KBS UI design	**3.58**	**2.60**	**3.84**	**2.57**	**3.12**

Table 3: Results of the comparative user study regarding five selected KBS patterns, ratings from 1 (very good) to 6 (very bad). Not explicitly listed are Q2 which concerned the acquired solution (mirrored in the success rate), and Q8 (resolution of the screen).

Table 3 summarizes the questionnaire-based results of the comparative study (rating scale: 1/very good – 6/very bad). The first major finding is a strong correlation between KB quality Q4 and each of KB utility Q1 (0.9968), KBS efficiency Q7 (0.9813), and perceived correctness of the result Q3 (0.9571), correlation coefficient given in parentheses. Further, KB quality Q4 correlates quite strong with the overall KBS success (0.8325); thus overall, the KB quality can be assumed one major influencing factor regarding the overall KBS perception. This in turn explains the bad overall results for both Interview variants, despite their way more positive rating in the expert usability assessment: Both variants used a qualitatively rather unfavorable statistical KB—confirmed strongly also by subjective remarks. Yet, regarding Strict Interview, at least the basic tendency of the expert assessment—which confirmed a highly intuitive overall UI/interaction—was confirmed, see Q6 and Q9.

Box Questionnaire obviously received the best ratings, closely followed by the Daily variant; along with provided subjective remarks this indicates, that the more structured presentation of the Box variant was favored over the compact Daily layout, thereby consenting with the basic expert assessment findings. Apart from underlining this tendency, however, subjective remarks specifically criticized the more space consuming presentation of Box Questionnaire and the general lack of structure in Daily; those comments also revealed more approval

for Daily regarding its ease of use, simplicity, and efficiency. Thus, we suspect an even further increase in the overall rating of Daily in case it is further enhanced regarding its presentation—including, e.g., a clearer distinction of question and answer items, a more prominent highlighting of answered items, and providing overall visual structure by indicating separators between question/answer lines.

Regarding Hierarchical Clarifier, the ratings may seem improvable; yet, this KBS addressed a highly complex KB from the domain of protection against unlawful dismissal, with an equally comprehensive problem description of the dismissal conditions, the correctness of which was to be rated by the KBS. Thus, an utility value of 2.68 and even the more a success rate of 88.31 % are particularly good results in the given context of a highly expertise domain but domain laymen users. Especially the descriptive and transparent style of Hierarchical Clarifier, mirroring the derived question/solution ratings directly in the UI may have supported that result; it most likely fostered the overall trustworthiness Q3 (compared to the Interview variants).

As a general important insight it excelled clearly, that the evaluation of a KBS UI always is inherently coupled with the applied KB and the respective problem to be solved.

4.3 Case Studies with pattern-driven KBS Development

Pattern-driven development along with the tool ProKEt already proved highly beneficial in actual projects. First, we noticed a strong support of the requirements engineering process. In the Mediastinitis project—where a documentation KBS for the structured input of operation data is realized, c.f., [3]—the patterns, and their ProKEt reference implementations, provided a visual and interactive means for gathering the user requirements more precisely and quickly. Thus, it was easy to experiment with several Questionnaire variants—two distinct Box layouts and one Daily layout—and to let the user formulate his requirements based on practically investigating the ProKEt pattern demos.

Another advantage is the fostered reuse of KBS solutions. In the EuraHS project, c.f. also [3], nearly the same constraints and conditions existed as in Mediastinitis. Thus, it was quickly agreed that a similar KBS solution would fit best in that case, too. There, the availability of the Questionnaire reference implementation in ProKEt drastically accelerated the initial setup of a first functional demo system—which was gradually refined, particularly regarding the KB, later, yet the project initiation itself was highly accelerated and eased.

Similarly, in the JuriSearch project, see [2]—aiming at providing clarification consultation modules for diverse legal topics—we could easily experiment with a Hierarchical Clarifier and a (preliminary) hybrid Clarification Interview variant regarding the most suiting solution.

5 Conclusion

In this paper, we motivated the benefits of pattern-driven development in the context of KBS. For practical support, we proposed a pattern specification frame-

work, based on tailored KBS (usability) criteria and a pattern template, and we introduced a collection of 10 fundamental KBS UI/interaction patterns. Further, we reported practical experiences with the approach: Using the tailored KBS engineering tool ProKEt for creating reference implementations of the patterns, evaluating five of them regarding their design and usability, and empirical experiences with pattern-driven KBS development from current projects.

Despite denoting an exciting approach for leveraging encompassing KBS development, there are several aspects worth investigating in the future: First, the general extension of the tool ProKEt as to entirely support those patterns that are specified theoretically only so far. Second, a thorough usability- and design-related base evaluation of the not yet assessed patterns. Third, an in-depth inquiry of one assumption that has emerged in the conducted study: That a structural enhancement of Daily Questionnaire may entail even better results compared to Box Questionnaire. Fourth, follow-up studies for investigating patterns in direct comparison for delimitating core characteristics even clearer—e.g., the pure Clarifier variants vs. one or both Clarifier Hybrids. Another goal is the actual application of further selected patterns in current (and potential future) projects; e.g., using Clarification Interview as a first-time user alternative for the Hierarchical Clarifier in the JuriSearch project.

References

1. Freiberg, M., Baumeister, J., Puppe, F.: Interaction Pattern Categories—Pragmatic Engineering of Knowledge-Based Systems. In: Proceedings of the 6th Workshop on Knowledge Engineering and Software Engineering (KESE-2010) at the 33rd German Conference on Artificial Intelligence (2010)
2. Freiberg, M., Puppe, F.: 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 (2012)
3. Freiberg, M., Striffler, A., Puppe, F.: Extensible prototyping for pragmatic engineering of knowledge-based systems. Expert Systems with Applications 39(11), 10177 – 10190 (2012)
4. Lidwell, W., Holden, K., Butler, J.: Universal Principles of Design. Rockport Publishers Inc., 2nd edn. (2010)
5. Nielsen, J.: Heuristic Evaluation. In: Nielsen, J., Mack, R.L. (eds.) Usability Inspection Methods, pp. 25–62. John Wiley & Sons, New York (1994)
6. Puppe, F.: Knowledge Formalization Patterns. In: Proceedings of PKAW 2000, Sydney Australia (2000)
7. Rogers, Y.: Interaction Design—beyond human-computer interaction. John Wiley & Sons, Ltd, 3rd edn. (2011)
8. Schmettow, M.: User interaction design patterns for information retrieval systems (2006)
9. Tidwell, J.: Designing Interfaces — Patterns for Effective Interaction Design. O'Reilly Media Inc. (2006)
10. Wharton, C., Rieman, J., Lewis, C., Polson, P.: Usability inspection methods. chap. The Cognitive Walkthrough Method: A Practitioner's Guide, pp. 105–140. John Wiley & Sons, Inc., New York, NY, USA (1994)