Subgroup Mining for Interactive Knowledge Refinement

Martin Atzmueller[1], Joachim Baumeister[1], Achim Hemsing[2], Ernst-Jürgen Richter[2], and Frank Puppe[1]

[1] Department of Computer Science, University of Würzburg, 97074 Würzburg, Germany
{atzmueller, baumeister, puppe}@informatik.uni-wuerzburg.de
[2] Department of Prosthodontics, University of Würzburg, 97070 Würzburg, Germany
{hemsing_a, richter_e}@klinik.uni-wuerzburg.de

Abstract. When knowledge systems are deployed into a real-world application, then the maintenance of the knowledge is a crucial success factor. In the past, some approaches for the automatic refinement of knowledge bases have been proposed. Many only provide limited control during the modification and refinement process, and often assumptions about the correctness of the knowledge base and case base are made. However, such assumptions do not necessarily hold for real-world applications. In this paper, we present a novel interactive approach for the user-guided refinement of knowledge bases. Subgroup mining methods are used to discover local patterns that describe factors potentially causing incorrect behavior of the knowledge system. We provide a case study of the presented approach with a fielded system in the medical domain.

1 Introduction

In the medical domain knowledge systems are commonly built manually by domain specialists. When such systems are deployed into a real-world application, then often the correctness needs to be improved according to the practical requirements. In the past, many approaches for the automatic refinement of knowledge bases have been proposed [1–4]. However, such methods make two important assumptions that do not necessarily hold in a real-world setting. The first assumption states that the considered knowledge base is mainly correct and only requires minor modifications in the refinement step, i.e., the *tweak assumption*. This assumption does not hold, if the development of the knowledge base is in an earlier stage, and if corrections or extensions are still necessary. As the second assumption a collection of correctly solved test cases is expected. These cases are used by the methods for identifying *guilty* (faulty) elements in the knowledge base, that are the target for refinement in a subsequent step. Unfortunately, this assumption is not valid in our setting since the available cases were manually entered. Although the user is guided by an adaptive dialog during the case acquisition phase, and consistency checks are applied, we frequently experienced falsely entered findings in our case study.

In this paper, we present a novel approach for the user-guided refinement of knowledge bases. The proposed method supports the user to perform the correct refinements in an interactive process. This is especially important if the formalized knowledge is still incomplete, i.e., no tweak assumption for the underlying knowledge base can be made. In such circumstances, extensions and not only modifications of the knowledge base are necessary. Furthermore, if manually acquired case bases are used to refine knowledge systems, then the applied case base may contain incorrectly solved cases, e.g., due to incorrectly entered findings or solutions. Additionally, it is possible that automatic methods overfit the learned (refinement) knowledge by over-generalization or over-specialization. This problem is increased by the presence of incorrectly solved cases. Then, automatic refinements may not be acceptable for the expert.

In the presented approach subgroup mining methods are used to discover local patterns that describe factors potentially causing incorrect behavior of the knowledge system. It is important that no global refinement model of the knowledge base is generated but refinement operators are proposed based on a local model. The proposed method keeps the domain specialist in control of all steps of the refinement process. The user is supported by visualization techniques to easily interpret the (intermediate) results.

The rest of the paper is organized as follows: In Section 2 we introduce subgroup mining and its application for the refinement task. In Section 3, we present the subgroup driven interactive refinement process: we discuss the refinement steps, a visualization technique and related work. Finally, we provide a case study of the presented approach with a fielded system in the medical domain in Section 4. A summary of the paper is given in Section 5.

2 Subgroup Mining

In this section, we first introduce our knowledge representation, and we describe the basics of the subgroup mining approach. After that, we introduce the adaptation of subgroup mining to the interactive refinement process.

General Definitions Let Ω_D be the set of all diagnoses and Ω_A the set of all attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of attribute values is defined. Furthermore, we assume $\mathcal{V}_A$ to be the (universal) set of attribute values (*findings*, *observations*) of the form $(a : v)$, where $a \in \Omega_A$ is an attribute and $v \in dom(a)$ is an assignable value. For each diagnosis $d \in \Omega_D$ we define a (boolean) range

$$dom(d)\text{: } \forall d \in \Omega_D : dom(d) = \{established, not\ established\}.$$

Let CB be the case base containing all available cases. A case $c \in CB$ is defined as a tuple $c = (\mathcal{V}_c, \mathcal{D}_c)$, where $\mathcal{V}_c \subseteq \mathcal{V}_A$ is the set of attribute values observed in the case c. The set $\mathcal{D}_c \subseteq \Omega_D$ is the set of diagnoses describing the *solution* of this case. The occurrence of a diagnosis d in a case c, indicates the value $established$. The value $not\ established$ does not occur in our case base. Thus, $\mathcal{V}_F = \mathcal{V}_A \cup \Omega_D$ denotes the (universal) set of all possible "generalized" attribute values of the case base CB.

A diagnosis $d \in \Omega_D$ is derived using (heuristic) rules. A rule r can be considered as a triple $\big(cond(r), conf(r), d\big)$, where $cond(r)$ is the condition of the rules, $conf(r)$ is the confirmation strength (points), and $d \in \Omega_D$ is a diagnosis. Thus a rule $r = cond(r) \rightarrow d, conf(r)$ is used to derive the diagnosis d, where the rule condition $cond(r)$ contains conjunctions and/or disjunctions of (negated) generalized findings $f_i \in \mathcal{V}_F$. The state of a diagnosis is gradually inferred by summing all the confirmation strengths (points) of the rules that have fired; if the sum is greater than a specific threshold value, then the diagnosis is assumed to be established.

2.1 Basic Subgroup Mining

Subgroup mining [5, 6] is a method to discover "interesting" subgroups of cases, e.g., in the domain of dental medicine the subgroup "teeth with a strong attachmentloss and an increased degree of tooth lax" has a significantly higher share of "extracted teeth" than the total population. The main application areas of subgroup mining are exploration and descriptive induction: subgroups are described by relations between independent (explaining) variables and a dependent (target) variable rated by a certain interestingness measure.

A subgroup mining task mainly relies on the following four main properties: the target variable, the subgroup description language, the quality function, and the search strategy. We will focus on binary target variables.

The description language specifies the individuals from the reference population belonging to the subgroup.

Definition 1 (Subgroup Description). *A subgroup description $sd = \{e_i\}$ consists of a set of selection expressions (selectors) $e_i = (a_i, V_i)$ that are selections on domains of attributes, where $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. A subgroup description is defined as the conjunction of its contained selection expressions. We define Ω_{sd} as the set of all possible subgroup descriptions.*

A quality function measures the interestingness of the subgroup. Several quality functions were proposed, for example in [6, 7].

Definition 2 (Quality Function). *A quality function*

$$q : \Omega_{sd} \times \mathcal{V}_F \rightarrow R$$

evaluates a subgroup description $sd \in \Omega_{sd}$ given a target variable $t \in \mathcal{V}_F$. It is used by the search method to rank the discovered subgroups during search.

For binary target variables, examples for quality functions are given by

$$q_{BT} = \frac{p - p_0}{\sqrt{p_0 \cdot (1 - p_0)}} \sqrt{n} \sqrt{\frac{N}{N - n}}\,, \qquad q_{TP} = \frac{pn}{(1 - p)n + g}\,,$$

where p is the relative frequency of the target variable in the subgroup, p_0 is the relative frequency of the target variable in the total population, $N = |CB|$ is the size of the total population, and n denotes the size of the subgroup. For quality function q_{TP} the generalization parameter g trades of the number of true positives (pn) vs. the number of false positives $\big((1 - p)n\big)$. For a low value of g fewer false positives are tolerated.

Considering an automatic subgroup mining approach an efficient search strategy is necessary, since the search space is exponential concerning all possible selection expressions. Commonly, a beam search strategy is used because of its efficiency. We use a modified beam search strategy, where a subgroup description can be selected as the initial value for the beam. Beam search adds selection expressions to the k best subgroup descriptions in each iteration. Iteration stops, if the quality as evaluated by the quality function q does not improve any further.

For the characterization of the discovered subgroups we have two alternatives: Besides the principal factors contained in the subgroup description there are also supporting factors. These are generalized findings $supp \subseteq \mathcal{V}_F$ contained in the subgroup, which are characteristic for the subgroup, i.e., the value distributions of their corresponding attributes (supporting attributes) differ significantly comparing two populations: the true positive cases contained in the subgroup and non-target class cases contained in the total population. In addition to the principal factors the supporting factors can also be used to statistically characterize a discovered subgroup, as described, e.g. in [8].

2.2 Subgroup Mining for the Refinement Task

For subgroup mining we consider a binary target variable corresponding to a diagnosis d, that is true (established) for incorrectly solved cases. Then, we try to identify subgroups with a high share of this "error" target variable. However, we need to distinguish different *error analysis states* relating to the measures *false positives* $FP_d(CB)$, *false negatives* $FN_d(CB)$, and the total error $ERR_d(CB)$:

$$
\begin{aligned}
FP_d(CB) &= \left|\{c \,|\, \mathcal{C}D_c \neq \emptyset \,\wedge\, d \in \mathcal{S}D_c \,\wedge\, d \notin \mathcal{C}D_c\}\right|, \\
FN_d(CB) &= \left|\{c \,|\, \mathcal{C}D_c \neq \emptyset \,\wedge\, d \notin \mathcal{S}D_c \,\wedge\, d \in \mathcal{C}D_c\}\right|, \\
ERR_d(CB) &= \left|\{c \,|\, \mathcal{C}D_c \neq \emptyset \,\wedge\, \mathcal{S}D_c \neq \mathcal{C}D_c\}\right|,
\end{aligned}
$$

where $\mathcal{C}D_c$ are the *correct diagnoses* of the case c, and $\mathcal{S}D_c$ are the diagnoses derived by the system. It is easy to see that we want to minimize the measures for the (general) refinement task, while we want to maximize the measures for the discovered subgroups that are then used as candidates for refinement.

To identify the "potential faulty factors" PFF we consider the subgroup descriptions of the discovered subgroups containing a high share of falsely solved cases. Then there are two options: the interesting factors are always the *principal factors* describing the subgroup, i.e., the attribute values contained in the subgroup description. Additionally, also the *supporting factors* of the subgroup can be faulty factors, since their distribution differs significantly considering the incorrectly and correctly solved cases. Then, the potential faulty factors PFF are defined as follows:

$$PFF = \{\, f \mid f \text{ is principal or supporting factor} \,\}.$$

For the refinement task we also apply static test knowledge, i.e., immutable validation constraints, to detect inconsistent behavior of the knowledge system. These constraints are provided by the domain specialist as subgroups for which a specific diagnosis should always be derived. Then, by assessing the distribution of the diagnoses contained in these subgroups, we can validate the state of the knowledge base or the case base directly. Furthermore, after a refinement step has been performed, the test knowledge is always checked again, in order to exclude modifications which degrade the performance of the system. Examples of the static test knowledge are given in the case study in Section 4.

3 The Subgroup-Driven Interactive Refinement Process

In this section, we introduce the process for interactive knowledge refinement and its characteristics. We present the visualization method which provides an easy interpretation of intermediate results. Finally, we discuss related work.

3.1 Subgroup Mining for Interactive Knowledge Refinement – Process Model

For the interactive refinement process we apply the subgroup mining method to discover local patterns describing a set of incorrectly solved cases. We aim to discover subgroup cases with a high share of incorrectly solved cases. The incremental process depicted in Figure 1 mainly consists of six steps:

1. Consider a diagnosis $d \in \Omega_D$, and select an analysis state $e \in \{FP_d, FN_d, ERR_d\}$.
2. A set of subgroups $\mathcal{SGS}_e$ is mined, either interactively by the domain specialist, or automatically by the system. Then, for each subgroup $\mathcal{SG}_i \in \mathcal{SGS}_e$ a set of *potential faulty factors* PFF_i contained in $\mathcal{SG}_i$ is retrieved.
3. The subgroup descriptions/factors are interpreted by the domain specialist.
4. Based on the analysis of the potential faulty factors *guilty* (faulty) elements in the knowledge base or the case base are identified, and appropriate modification steps are applied. Then, the solutions of each case in the case base are recomputed.
5. The (changed) state of the system is assessed: the analysis measure e is checked for improvements; similarly the immutable validation constraints, if available, are tested whether they still indicate a valid state.
6. If necessary, restart the process.

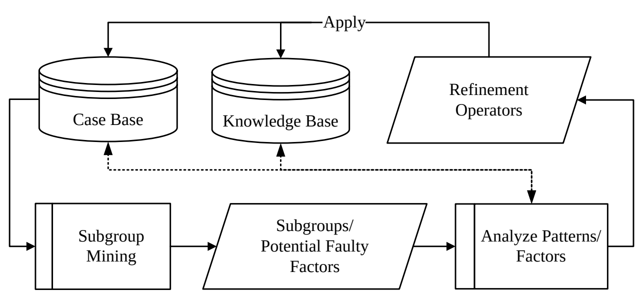

Fig. 1. Process Model: Subgroup Mining for Interactive Knowledge Refinement

Refinement operators can either modify the knowledge base or the applied case base. The knowledge base is usually adapted to fit the available correct cases. The case base is adapted, if particular cases are either wrong or they denote an extraordinary, exceptional state, which should not be modelled in the knowledge base. For the different refinement operators we need to distinguish two cases: if the expert decides that the subgroup descriptions are valid, i.e., they are reasonable, then probably the knowledge base needs to be corrected. Otherwise, if the subgroup descriptions, i.e., the combination of factors are not meaningful, then this can imply that the contained cases need corrections.

However, these "doubtful" subgroups could also be caused by random correlations in the case base. In this case, the expert needs to manually assess the subgroups and cases in detail. In summary, the following refinements can be performed:

– **Adapt/modify rules:** generalize or specialize conditions and/or actions. This action is often appropriate if only one selector is contained in the subgroup, and if the subgroup is assessed to be valid.
– **Extend knowledge:** add missing relations to the knowledge base. This operator is often applicable when the subgroup description consists of more than one selector, and if the dependencies between the selectors are meaningful.
– **Fix case:** correct the solution of a single case, or correct the findings of a case, if the domain specialist concludes in a detailed case analysis that the case has been labeled with the wrong solution.
– **Exclude case:** exclude a case completely from the analysis. If the behavior modeled by the case cannot be explained by factors inherent in the knowledge base, e.g., by external decisions, then the case should be removed. This happened in our case study only for a low number of cases.

Examples of the application of the refinement operators are given in Section 4.

3.2 Visualizing Subgroups and Interesting Factors

If the user is not supported by visualization techniques, then an interactive refinement approach typically is not tractable, since the refinement space is usually large. Therefore, we provide visualization methods that enable the user to browse the space of subgroup hypotheses, while testing the hypotheses interactively. This process can also be supported by automatic subgroup mining methods that provide an initial starting point for further interaction. Additionally, visualization techniques simplify the interpretation of the subgroup mining results. Furthermore, they should guide the user to the right direction for refinement.

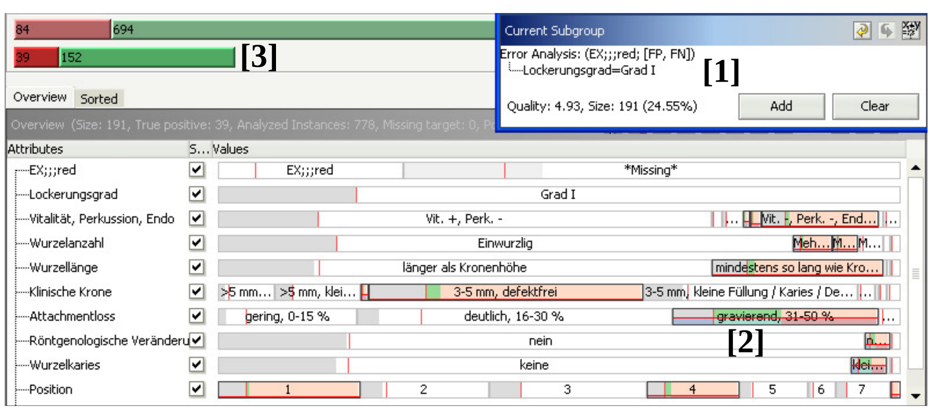

Fig. 2. Visualizing Subgroups and Interesting Factors (in German)

An examplary visualization is shown in Figure 2 where the distributions of several factors are given. The subgroup *toothlax = minor (Lockerungsgrad = Grad I)* (Annotation 1) is shown with 39 incorrectly solved cases and 152 correctly solved cases; the general population contains 84 incorrectly and 694 correctly solved cases (Annotation 3).

The rows in the table below the subgroup show the value distributions of the other attributes. Labels with a large "dark-gray" (green) sub-label, or a (red) horizontal bar that is close to the top, indicate "interesting" attribute values: the width/height, respectively, of these sub-bars indicates the *improvement* of the target share, if the respective attribute value (selector) would be added to the current subgroup, resulting in a virtual *future* subgroup. The horizontal line indicates the *relative improvement* of the target share: if the line is in the middle of the attribute value bar, then the target share in the future subgroup improves by 50%. In addition to the improvement of the target share of the current subgroup, the (potential) reduction of the subgroup size shown by the width of a attribute value cell also needs to be taken into account. In the example visualization the cell *attachmentloss = strong (Attachmentloss = gravierend, 31-50%)* is the best one considering its size, and also the target share (Annotation 2). Another interesting factor could be *Position = 4* which, however, can be regarded as a random finding.

In this visualization the user is able to inspect different subgroups directly by one click on the corresponding cells. All elements, i.e., subgroups, rules, and cases, can be browsed directly by one click, and changes can be assessed immediately. The changes are also intuitively reflected by the size of the bars (Annotation 3). Therefore, the user-guided integrated method provides direct interaction and instant feedback to the user.

3.3 Discussion

In the past, various approaches for knowledge refinement were proposed, e.g. [1]. More recently, Knauf et al. [4] presented a refinement approach embedded in a complete validation methodology. Carbonara and Sleeman [2] describe an efficient method for selecting effective refinements, and Boswell and Craw [3] introduce a set of general refinement operators that are applicable in various application domains and that can be used within different problem-solving tasks. All these approaches are classified as *automatic refinement techniques* modifying rule based knowledge. The modifications are motivated by a previous analysis step performing a *blame allocation*, i.e., identifying faulty knowledge. Then, alternative strategies are applied in order to automatically generate possible and select suitable refinements of the knowledge base.

However, all automatic methods make the *tweak assumption* [2], which implies that the knowledge base is almost valid and only small improvements need to be performed. In our application scenario the validity of the knowledge base was quite poor (about 86% accuracy) and therefore no tweak assumption could be made. In contrast, we expected that important rules were missing and that we have to acquire additional knowledge during the process. For this reason, we decided to choose a mixed refinement/elicitation process, which emphasizes the interactive analysis and modification of the implemented rules based on found subgroup patterns. Similarly, Carbonara and Sleeman [2] use an inductive approach for generating new rules using the available cases. Additionally, in our application we cannot expect that all cases contain the correct solution, and thus a thorough analysis of the cases within the process was also necessary. In contrast, automatic approaches mainly do assume a correct case base.

4 Case Study

In this section, we introduce the applied medical domain and we present practical experiences with the described approach.

4.1 The Prothetic System

The case study was implemented with a consultation and documentation system for dental findings regarding any kind of prosthetic appliance. The system was developed by the department of prosthodontics at the Würzburg University Hospital in cooperation with the department of computer science VI of the University of Würzburg. The domain specialists used the knowledge system D3 [9] to implement the knowledge base.
The systems aims to decide about a diagnostic plan using the clinical findings to accommodate the patient with denture. In the first level the system proposes the teeth that could be conserved and the teeth that should be extracted. The cases contain always the standard findings acquired in the first consultation with the patient, and additional findings from x-ray examinations, e.g., abnormal x-ray findings (apical, periradicular), grade of tooth lax, endodontic state (root filling, pulp vitality), root quantity, root length, crown length, level of attachment loss, root caries, tooth angulation and elongation/extrusion. The cases are manually entered by the examiners using an interactive dialog. For a given tooth all findings are stored in a single case in a data base.
In the knowledge base each finding obtains a point score depending on its quality. The outcome of the addition of the single scores is the dental score of the examined tooth. If the total dental score is less or equal than 40 points, then the tooth should be conserved. If the dental score is greater than 40 points, then the tooth has to be extracted (*EX*).
The system tries to support the dentist with time-efficient planning of patients' denture. Additionally, it should increase the efficiency of clinical work by chairside taking findings that are immediately translated to a prosthodontic therapy decision. In the future, it is also envisioned to use the diagnostic decision tool as a knowledge-based system for dental student education in order to train the ensuing diagnostic work up. Then, students can learn recognition and interpretation of symptoms and clinical findings by comparing their diagnosed solutions with the derived solutions of the system.

4.2 Results

To assess the quality of the system we compared the results of the system with the solutions of a domain specialist, both using the same set of findings. The initial case base contained 802 cases. 24 cases were removed from the case base, because the corresponding teeth had been extracted by prosthodontic reasons during planning denture. Although these teeth had a better dental score their extraction (*EX*) was decided, e.g., to prevent irregular construction in denture which can cause problems in future. Finally, the applied case base contained 778 cases corresponding to 778 examined teeth. We investigated the diagnosis corresponding to tooth extraction/non extraction. Considering this diagnosis the case base contained 108 false positive and 670 correct cases without any refinement of the knowledge base.

First subgroup mining efforts turned out unexpected subgroups with a very high share of incorrectly solved cases. However, some subgroup descriptions were very difficult to interpret by the domain specialist, since they contained finding combinations that should establish the diagnosis *EX* categorically. Therefore, the domain specialist provided immutable test knowledge represented as a set of synthetically generated test cases. Examples are shown in Table 1. The given subgroup descriptions indicate certain knowledge, when the diagnosis *EX* should be categorically established.

No.	*Subgroup Description (findings)*	*Diagnosis*
1	tooth lax = medium $\wedge$ attachmentloss = strong	EX
2	attachmentloss = very strong	EX
3	tooth lax = strong $\wedge$ root quantity = 3	EX
4	tooth lax = strong $\wedge$ root caries = deep caries	EX

Table 1. Examples for immutable test knowledge

Using these subgroups the domain specialist was immediately able to locate incorrect cases, due to problems concerning data acquisition, i.e., noise in cases. Either the cases contained a false solution or incorrect findings. In total, 19 cases were corrected: 16 contained false diagnoses, and 3 contained incorrect case descriptions (findings).
Using the refined case base further analysis by automatic subgroup mining turned up several subgroups which were assessed as *dubious* by the expert, e.g., a subgroup described by *tooth lax = medium* $\wedge$ *tooth position = 2* $\wedge$ *tooth quadrant = 2*. However, these subgroups had a high share of incorrectly solved cases. Since the combined potential faulty factors did not indicate anything particular, the domain specialist checked the contained cases. It turned out, that the *false positives* and *false negatives* had a high share of incorrect case descriptions and incorrectly assigned solutions. In total, further 12 cases were fixed.

No.	*Subgroup Description*	*Diagnosis*	*Points*
1	abnormal x-ray = only apical	EX	10 $\longrightarrow$ 5
2	tooth lax = medium $\wedge$ root length = longer than crown length	EX	-20
3	tooth lax = minor $\wedge$ attachmentloss = strong	EX	-20

Table 2. Discovered subgroups indicating a knowledge base refinement

Using subgroup mining for the refinement of the knowledge system we managed to improve the knowledge base by reducing the number of incorrectly solved cases down to 54 cases: the domain specialist assessed several subgroups mined by the system as significant, which were then used for knowledge base refinement. We modified and added several rules, examples are given in Table 2. Subgroup description #1 is an example for a simple modification. For *abnormal x-ray = only apical* we modified the score, such that the rule only contributes 5 points. The last two subgroup descriptions the corresponding rules exemplify two general mechanisms: in rule #2 the condition *root length = longer than crown length* counts as negative for extraction, and relativizes the factor *tooth lax = medium* which is positive for extraction. Such an interaction can also work the other way round, i.e., when a positive factor influences a negative one. Then, for extraction, we would have to add points, e.g., for *tooth lax = medium* and *attachmentloss = minor*. For subgroup description #3 the selectors *tooth lax = minor* and *attachment-*

loss = strong are both positive for extraction, but since they are assessed independently in the rule base they should not be over-emphasized by being counted twice. Therefore, the score points of the corresponding rules were decreased. The system also discovered some subgroups that could not be interpreted by the experts afterwards and thus were ignored, e.g., *tooth lax = medium* $\wedge$ *root quantity = 1*. In summary, we managed to reduce the number of incorrectly solved cases from 108 to 54 by 50%. In consequence, we increased the precision of the knowledge base from 86% to 93%. The method was very well accepted by the domain specialist, who was able to directly inspect and change the subgroups and cases by himself.

5 Summary and Future Work

In this paper, we presented a novel method using subgroup mining for interactive knowledge refinement. We introduced the application of the mining method, and proposed a process model for the refinement task. Furthermore, we described the identification of potential faulty factors and discussed the applicable refinement operators. We also motivated how the user of the process is supported by visualization techniques that guide the interactive refinement process. In a case study using cases from a fielded medical system we demonstrated the application and the benefit of the proposed techniques.
In the future, we plan to investigate further visualization techniques to support the user during the refinement process. Additionally, a semi-automatic refinement method that is adapted to the used knowledge representation (rules with point scores) is another interesting issue to consider.

References

1. Ginsberg, A.: Automatic Refinement of Expert System Knowledge Bases. Morgan Kaufmann (1988)
2. Carbonara, L., Sleeman, D.: Effective and Efficient Knowledge Base Refinement. Machine Learning **37** (1999) 143–181
3. Boswell, R., Craw, S.: Organizing Knowledge Refinement Operators. In: Validation and Verification of Knowledge Based Systems. Kluwer, Oslo, Norway (1999) 149–161
4. Knauf, R., Philippow, I., Jantke, K.P., Gonzalez, A., Salecker, D.: System Refinement in Practice – Using a Formal Method to Modify Real-Life Knowledge. In: Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2002), AAAI Press (2002)
5. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In Komorowski, J., Zytkow, J., eds.: Proc. First European Symposion on Principles of Data Mining and Knowledge Discovery (PKDD-97), Berlin, Springer Verlag (1997) 78–87
6. Klösgen, W.: Subgroup Discovery. Chapter 16.3. In: Handbook of Data Mining and Knowledge Discovery. Oxford University Press, New York (2002)
7. Gamberger, D., Lavrac, N., Krstacic, G.: Active Subgroup Mining: a Case Study in Coronary Heart Disease Risk Group Detection. Artificial Intelligence in Medicine **28** (2003) 27–57
8. Gamberger, D., Lavrac, N.: Expert-Guided Subgroup Discovery: Methodology and Application. Journal of Artificial Intelligence Research **17** (2002) 501–527
9. Puppe, F.: Knowledge Reuse among Diagnostic Problem-Solving Methods in the Shell-Kit D3. Intl. Journal of Human-Computer Studies **49** (1998) 627–649