Prototypical Cases in Medical CBR Systems

Rainer Schmidt[1)], Olga Vorobieva[1+2)], Tina Waligora[1)]

[1)] Institut für Medizinische Informatik und Biometrie, Universität Rostock, Germany
rainer.schmidt@medizin.uni-rostock.de
[2)] Sechenov Institute of Evolutionary Physiology and Biochemistry, St.Petersburg, Russia

Abstract. Already in the early stages of Case-Based Reasoning prototypes were considered as an interesting technique to structure the case base and to fill the knowledge gap between single cases and general knowledge. Unfortunately, later on prototypes never became a hot topic within the CBR community. However, for medical applications they have been used rather regularly, because they correspond to the reasoning of doctors in a natural way. In this paper, we illustrate the role of prototypes by application programs, which cover all typical medical tasks: diagnosis, therapy, and course analysis.

Keywords: Prototypical Cases, Medical CBR Systems, Dysmorphic Syndromes, Kidney Function, Endocrine Therapy Support

1 Introduction

Cases are the most specialised form of knowledge representation. The knowledge of physicians consists of general knowledge they have read in medical books and of their experiences in the form of cases they have treated themselves or colleagues have told them about. Not all cases are of the same importance. Some are typical while others are rather exceptional, e.g. a paediatrician does not remember all his patients with a diagnosis of measles, but maybe those with serious complications or those where his measles diagnosis was surprisingly wrong. Doctors consider differences between their current patient and typical or known exceptional cases.

We believe that medical Case-Based Reasoning (CBR) systems should take the reasoning of doctors into account [1]. Such systems should not only consist of general medical domain knowledge plus a flat case base, but the case base should be structured by typical case generalisations called prototypes [2].

Though the use of prototypes had been early introduced in the CBR community [3, 4], their use is still rather seldom. Later on it fell into oblivion and was brought up again by Maximini et al. in the form of generalised cases [5]. Their notion of generalised cases is similar but not identical to our idea of prototypes. While generalised cases are general or abstract in contrast to concrete cases, prototypes contain the typical features of a set of cases.

However, since doctors reason with typical cases anyway, in medical CBR systems prototypes are a rather common knowledge form, they are used in a variety of applications, e.g. for diabetes [6], for eating disorders [7], and for pulmonology [8]. Prototypical images that can be transformed after certain image processing steps into prototypes are used for the diagnosis of medical images [9].

Recently Isabelle Bichindaritz [10] has presented her experience with prototypes for medical CBR systems. However, she focuses on their role for knowledge maintenance, which is an interesting topic but it lays beyond the scope of this paper.

A Prototype is generalised from a set of single cases. The cases in this set are very similar to each other or they belong together in some other specific way and form a sort of class. For example, in a diagnostic system all patients diagnosed as measles patients might be grouped together. Usually, prototypes have the same structure as cases but have less and more general features, namely just the typical ones. Sometimes prototypes are defined by medical experts, sometimes they can be found in literature (e.g. the typical symptoms for measles), and sometimes they are computed. The use of case oriented generalised knowledge presents the opportunity to structure case bases. Cases can be clustered into groups, prototypical diseases or schema. Clancey [11] distinguishes between prototypes that represent specific expressions of diseases or therapies and schema that contain essential features of diseases or therapies. As Selz [12] characterises a schema as a description of an entity where at least one

part remains vague, the distinction between prototypes and schema seems to be fluid. We only use the term prototype and refer to a hierarchy of prototypes where the most general prototypes that contain the most common features are situated on top and the most specific ones are placed at the bottom. This notion of prototypes differs from the usual notion of classes and clusters [13] in many ways. Prototypes are not the result of a classification process. Whether a case belongs to a prototype, is determined by its features or defined by an expert. There may be a hierarchy of prototypes but there are no relations (similarity, is-a and so on), and the set of cases belonging to a prototype is not represented by its most representative case but by the prototype. The main purpose of such generalised knowledge is to guide the retrieval and sometimes to decrease the amount of storage by deleting redundant cases. In domains with rather weak domain theories another advantage of case-oriented techniques is their ability to learn from cases. Only gathering new cases may improve the systems ability to find suitable similar cases for current problems, but it does not elicit the intrinsic knowledge of the stored cases. To learn the knowledge contained in cases a generalisation process is necessary. Generally speaking, prototypes fill the knowledge gap between the specificity of single cases and abstract knowledge usually expressed as rules.

In this paper we present three systems we developed during the last years and focus on the role of prototypes within them. A more detailed presentation on this topic can be found in [14]. We start with a prototype-based system for diagnosis of dysmorphic syndromes. Subsequently we present a system for course analysis and prognosis of the kidney function and finally we present ISOR, a system that deals with therapeutic problems in the endocrine domain.

2 Prototype-based Diagnosis of Dysmorphic Syndromes

In this application, retrieval does not search for former single cases but only for prototypes. Each prototype represents and characterises one specific diagnosis. We assume that this idea is rather typical for diagnostic tasks, because it seems to be reasonable to search for a general description of a disease instead of searching for single patients.

When a child is born with dysmorphic features or with multiple congenital malformations or if mental retardation is observed at a later stage, finding the correct diagnosis is extremely important. Knowledge of the nature and the etiology of the disease enables the pediatrician to predict the patient's future course. So, an initial goal for medical specialists is to diagnose a patient to a recognised syndrome. Genetic counselling and a course of treatments may then be established.

A dysmorphic syndrome describes a morphological disorder and it is characterised by a combination of various symptoms, which form a pattern of morphologic defects. An example is Down Syndrome which can be described in terms of characteristic clinical and radiographic manifestations such as mental retardation, sloping forehead, a flat nose, short broad hands, and generally dwarfed physique [15].

The main problems of diagnosing dysmorphic syndromes are [16]:

- more than 200 syndromes are known,
- many cases remain undiagnosed with respect to known syndromes,
- usually many symptoms are used to describe a case (between 40 and 130),
- every dysmorphic syndrome is characterised by nearly as many symptoms.

Furthermore, knowledge about dysmorphic disorders is continuously modified, new cases are observed that cannot be diagnosed (there is even a journal that only publishes reports of newly observed interesting cases [17]), and sometimes even new syndromes are discovered. Usually, even experts of paediatric genetics only see a small count of dysmorphic syndromes during their lifetime.

So, we have developed a diagnostic system that uses a large case base. The starting point to build the case base was a large case collection of the paediatric genetics department of the University of Munich, which consists of nearly 2,000 cases and 229 prototypes. A prototype (prototypical case) represents a dysmorphic syndrome by its typical symptoms. Most of the dysmorphic syndromes are already known and have been defined in the literature. And nearly one third of the prototypes were determined by semi-automated knowledge acquisition, where an expert selected cases that should belong to the same syndrome and subsequently a prototype, characterised by the most frequent symptoms of his cases, was generated. To this database we have added rare dysmorphic syndromes, namely from "clinical dysmorphology" [17] and from the London dysmorphic database [18].

2.1 Prototypicality measures

In CBR usually cases are represented as attribute-value pairs. In medicine, especially in diagnostic applications, this is not always the case. Instead, often a list of symptoms describes a patient's disease. Sometimes these lists can be very long, and often their lengths are not fixed but vary with the patient. For dysmorphic syndromes usually between 40 and 130 symptoms are used to characterise a patient.

Furthermore, for dysmorphic syndromes it is unreasonable to search for single similar patients (and of course none of the systems mentioned above does so) but for more general prototypes that contain the typical features of a syndrome. To determine the most similar prototype for a given query patient instead of a similarity measure a prototypicality measure is required. An important difference is that for prototypes the list of symptoms is usually much shorter than for single cases.

The result should not be just the one and only most similar prototype, but a list of them – sorted according to their similarity. So, the usual CBR retrieval methods like indexing or nearest neighbour search are inappropriate. Instead, rather old measures for dissimilarities between concepts [19, 20] are appropriate.

As humans look upon cases as more typical for a query case as more features they have in common [20], distances between prototypes and cases usually mainly consider the shared features. In some experiments we have compared the measure proposed by Tversky [19] with a similar measure proposed by Mervis and Rosch [20]. The measure proposed by Tversky (2.1) counts the number of matching symptoms of the query patient (X) and a prototype (Y). Subsequently, two numbers are subtracted, namely the number of symptoms that are observed for the patient but are not used to characterise the prototype (X-Y), and secondly the number of symptoms that are used for the prototype but are not observed for the patient (Y-X). Finally, the result is normalised by the number of symptoms used for the prototype (Y). In (2.1) *f* is a general function, however in our application it is just a counting function. The prototypicality measure proposed by Rosch and Mervis [20] differs from Tversky's measure only in one point: the factor X-Y is not considered.

$$\mathbf{D\,(X,Y)} = \frac{\mathbf{f\,(X+Y) - f\,(X-Y) - f\,(Y-X)}}{\mathbf{f\,(Y)}} \tag{2.1}$$

2.2 The System

Our system process consists of three steps. At first the user has to select the symptoms that characterise a new patient. This selection is a long and very time consuming process, because we consider more than 800 symptoms. However, diagnosis of dysmorphic syndromes is not a task where the result is very urgent, but it usually requires thorough reasoning and subsequently a long-term therapy has to be started. In the second step, a prototypicality measure is sequentially applied on all prototypes (syndromes). Since the syndrome with maximal similarity is not always the right diagnosis, the 20 syndromes with best similarities are listed in a menu. In the third and final step, the user can optionally choose to apply adaptation rules on the syndromes. Such a rule states that a specific combination of symptoms favours or disfavours specific dysmorphic syndromes. However, how shall the adaptation rules alter the results? Our first idea was that the adaptation rules should increase or decrease the similarity scores for favoured and disfavoured syndromes. But the question is how. Of course no medical expert can determine values to manipulate the similarities by adaptation rules and any general value for favoured or disfavoured syndromes would be arbitrary. So, instead the result after applying adaptation rules is a menu that contains up to three lists (figure 1).

On top the favoured syndromes are depicted, then those neither favoured nor disfavoured, and at the bottom the disfavoured ones. Additionally, the user can get information about the specific rules that have been applied on a particular syndrome. Rule-6, that favours the Lenz-syndrome, looks like that

If medial diffuse hypoplast brows
and if prominent corpus-anthelicis
then Lenz-Syndrome is propable

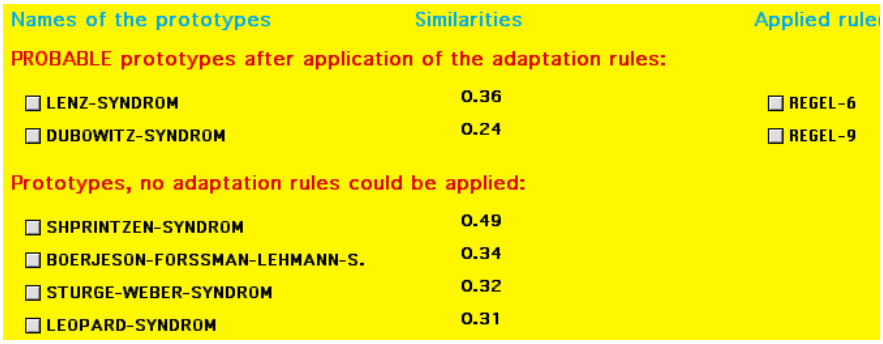

Fig. 1. Top part of the listed prototypes after additionally applying adaptation rules

In the example presented in figure 1, the correct diagnosis is Lenz-syndrome. The computation of the prototypicality measure provided Lenz-syndrome as the most similar but one syndrome. After application of adaptation rules, the ranking is not obvious. Two syndromes have been favoured, the more similar one is the right one. However, Dubowitz-syndrome is favoured too (by a completely different rule), because a specific combination of symptoms makes it probable, while other observed symptoms indicate a rather low similarity.

2.3 Results

Cases are difficult to diagnose when patients suffer from a very rare dysmorphic syndrome for which neither detailed information can be found in literature nor many cases are stored in our case base. This makes evaluation difficult. If test cases are randomly chosen, frequently observed cases resp. syndromes are frequently selected and the results will probably be fine, because these syndromes are well-known. However, the main idea of the system is to support diagnosis of rare syndromes. So, we have chosen our test cases randomly but under the condition that every syndrome can be chosen only once. For 100 cases we have compared the results obtained by both prototypicality measures. The measure proposed by Tversky provides slightly better results than the measure proposed by Mervis and Rosh. Subsequently, we additionally applied adaptation rules, first a set of just 10 rules, later on this set could be extended to 18 rules. The results are shown in table 1.

Table 1. Results

Right Syndrome	Rosch and Mervis	Tversky	Tversky (10 adaptation rules)	Tversky (18 adaptation rules)
on Top	29	40	42	44
among top 3	57	57	59	64
among top 10	76	69	71	73

The results may seem to be rather poor. However, diagnosis of dysmorphic syndromes is very difficult and usually needs further investigation, because often a couple of syndromes are very similar. The first step is to provide the doctor with information about probable syndromes, so that he or she gets an idea about which further investigations are appropriate. That means, the right diagnosis among the three most probable syndromes is already a good result.

Furthermore, the improvement mainly depends on the question how many syndromes involved by adaptation rules are among the test set. In our experiment this was the case only for 5 syndromes. Since some had been already diagnosed correctly without adaptation, there was just a small improvement.

It is obvious that with the number of acquired adaptation rules the quality of the program increases too. Unfortunately, the acquisition of these rules is very difficult and especially for very rare syndromes probably nearly impossible.

3 Prognosis of Kidney Function Courses

Up to 60% of the body mass of an adult person consists of water. The electrolytes dissolved in body water are of great importance for an adequate cell function. The human body tends to balance the fluid and electrolyte situation. But intensive care patients are often no longer able to maintain adequate fluid and electrolyte balances themselves due to impaired organ functions, e.g. renal failure, or medical treatment, e.g. parenteral nutrition of mechanically ventilated patients. Therefore physicians need objective criteria for the monitoring of fluid and electrolyte balances and for choosing therapeutic interventions as necessary.

At our ICU, physicians daily get a printed renal report from the monitoring system NIMON [21] which consists of 13 measured and 33 calculated parameters of those patients where renal function monitoring is applied. The interpretation of all reported parameters is quite complex and needs special knowledge of the renal physiology. The aim of our knowledge based system ICONS [22] is to give an automatic interpretation of the renal state to elicit impairments of the kidney function on time and to give early warnings against forthcoming kidney failures. That means, we need a time course analysis of many parameters without any well-defined standards. However, in the domain of fluid and electrolyte balance, there is no known prototypical approach in ICU settings and complete knowledge about the kidney function does not exist. Especially, knowledge about the behaviour of the various parameters over time is yet incomplete. So, we combined the idea of RÉSUMÉ [22] to abstract many parameters into one single parameter with the idea of Haimowitz and Kohane [24] to compare many parameters of current courses with well-known standards. Since well-known standards were not available, we used former similar cases instead.

3.1 Prognostic Method

The method that was developed for ICONS consists of three steps, namely a state abstraction, a temporal abstraction, and CBR retrieval.

State Abstraction. For the data abstraction we use states of the renal function, which determine states of increasing severity beginning with a normal renal function and ending with a renal failure. Based on the kidney function states, characterised by required and optional conditions for selected renal parameters, we first check the required conditions. For each state that satisfies the required conditions we calculate a similarity value concerning the optional conditions. We use a variation of Tversky's [19] measure of dissimilarity between concepts. If two or more states are under consideration, ICONS presents them to the user sorted according to their similarity values together with information about the satisfied and not satisfied optional conditions.

The user can accept or reject a presented state. When a suggested state has been rejected, ICONS selects another one. Finally, we determine the central state among the states (if more than one) the user has accepted. This central state is the closest one towards a kidney failure. Our intention is to find the state indicating the most profound impairment of the kidney function.

Temporal Abstraction. First, we have fixed five assessment definitions for the transition of the kidney function state of one day to the state of the respectively next day. These assessment definitions are related to the grade of renal impairment: steady, increasing, sharply increasing, decreasing, and sharply decreasing. These assessment definitions are used to determine the state transitions from one qualitative value to another. Based on these state transitions, we generate three trend descriptions. Two trend descriptions especially consider the current state transitions.

short-term trend:=	current state transition; Abbreviation: T1
medium-term trend:=	looks recursively back from the current state transition to the one before and unites them if they are both of the same direction or one of them has a “steady” assessment; Abbreviation: T2
long-term trend:=	characterises the considered course of at most seven days; Abbreviation: T3

For the long-term trend description we additionally introduced four new assessment definitions (alternating, oscillating, fluctuating, and nearly steady). If none of the five former assessments fits the complete considered course, we attempt to fit one of these four additional definitions.

Only if there are several courses with the same trend descriptions, we use a minor fourth trend description T4 to find the most similar among them. We assess the considered course by adding up the state transition values inversely weighted by the distances to the current day. Together with the current kidney function state these four trend descriptions form a course depiction, that abstracts the sequence of the kidney function states.

Figure 2 shows an example of a presentation of a query and a similar course. In the lower part of each course the (abbreviated) kidney function states are depicted. The upper part of each course shows the deduced trend descriptions. After another day with a "sharply reduced kidney function" the patient belonging to the similar course had a kidney failure. The physician may notice this as a warning and it is up to him to interpret it.

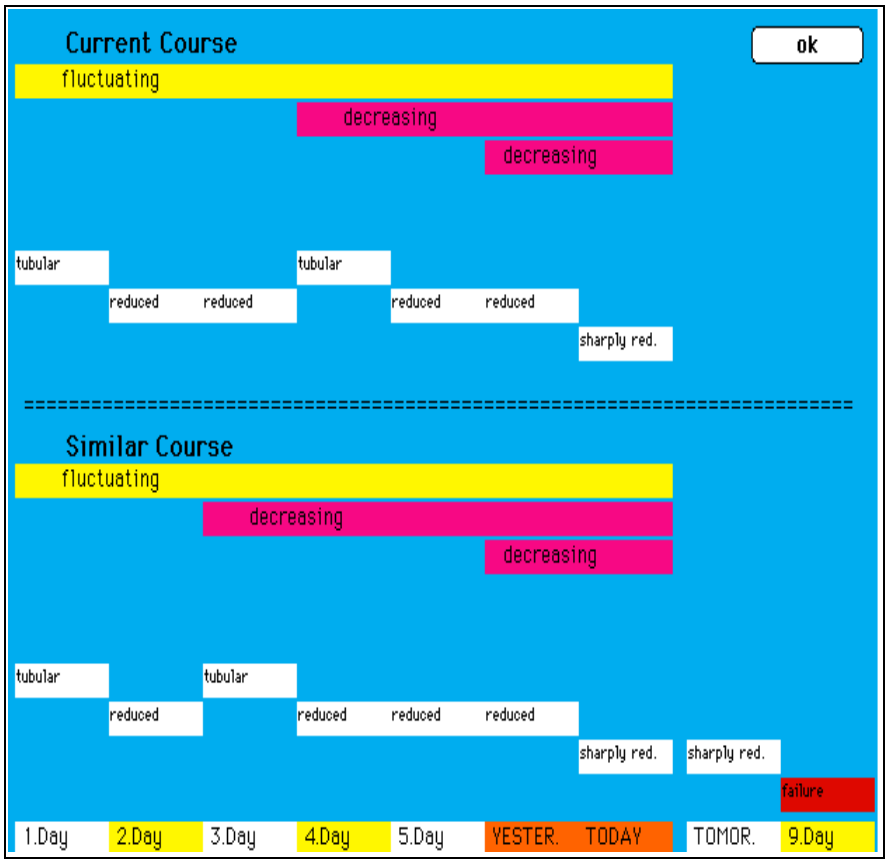

Fig. 2. Comparative presentation of a current and a similar course.

Retrieval. We use the parameters of the four trend descriptions and the current kidney function state to search for similar courses. As the aim is to develop an early warning system, we need a prognosis. For this reason and to avoid a sequential runtime search along the entire cases, we store a course of the previous seven days and a maximal projection of three days for each day a patient spent on the intensive care unit.

Since there are many different possible continuations for the same previous course, it is necessary to search for similar courses and for different projections. Therefore, we divided the search space into nine parts corresponding to the possible continuation directions. Each direction forms its own part of the search space. During the retrieval these parts are searched separately and each part may provide at most one similar case. The similar cases of these parts together are presented in the order of their computed similarity values.

Before the main retrieval, we search for a prototypical case (see Section 3.2) that matches most of the trend descriptions. Below this prototype the main retrieval starts. First we search with an activation algorithm concerning qualitative features. Only if two or more courses are selected in the same projection part, we use the sequential similarity measure TSCALE [25], which goes back to Tversky [19], concerning the quantitative features in a second step.

3.2 Learning a Tree of Prototypes

Prognosis of multi-parametric courses of the kidney function for ICU patients is a domain without a medical theory. Moreover, we can not expect such a theory to be formulated in the near future. So we attempt to learn prototypical course patterns. Therefore, knowledge in this domain is stored as a tree of generalised cases (prototypes) with three levels and a root node (figure 3).

Except for the root, where all not yet clustered courses are stored, each level corresponds to one of the trend descriptions T1, T2 or T3. As soon as enough courses that share another trend description are stored at a prototype, a new prototype with this trend is created. At a prototype at level 1, we cluster courses that share T1, at level 2, courses that share T1 and T2 and at level 3, courses that share all three trend descriptions T3. This can be done, because regarding their importance, the short-, medium- and long-term trend descriptions T1, T2 and T3 refer to hierarchically related time periods. T1 is more important than T2 and T3, and so forth.

The retrieval starts with a search for a prototype that has most of the trend descriptions in common with the query course. The search begins at the root node with a check for a prototype with the same short-term trend description T1. If such a prototype can be found, the search goes on below this prototype for a prototype that has the same trend descriptions T1 and T2, and so forth. If no prototype with a further trend in common can be found, we search for a course at the last accepted prototype.

If no prototype exists that has the same T1 as the query course, the search starts at the root node, where links to all courses in the case base exist.

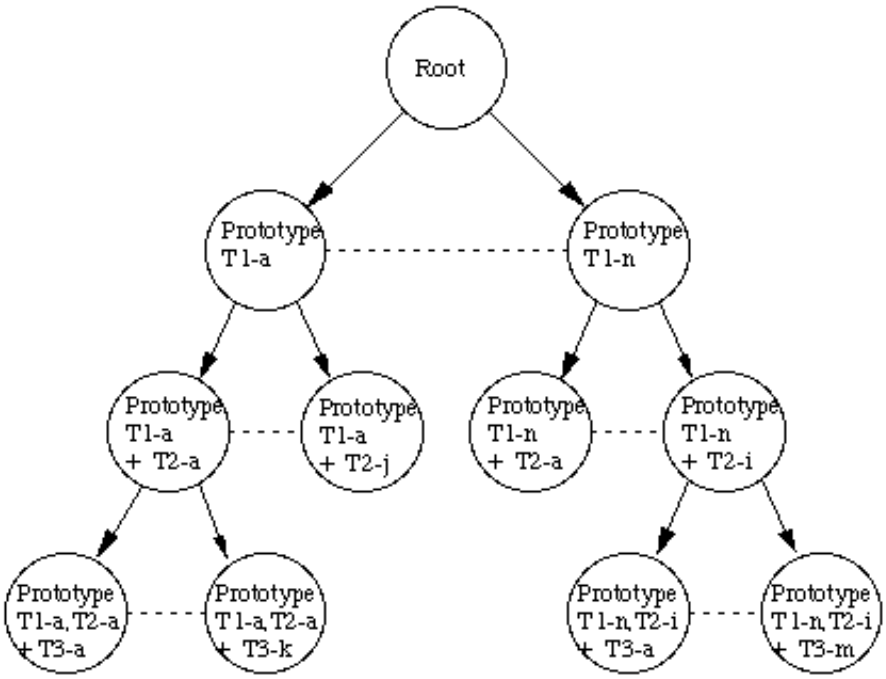

Fig. 3. Prototype architecture for the trend descriptions T1, T2, and T3.

4 ISOR

ISOR is a Case-Based Reasoning system for long-term therapy support in the endocrine domain [26]. It performs typical therapeutic tasks, such as computing initial therapies, initial dose recommendations, and dose updates. Apart from these tasks ISOR deals especially with situations where therapies become ineffective. Causes for inefficacy have to be found and better therapy recommendations should be computed. In addition to the typical CBR knowledge, namely former already solved cases, ISOR uses further knowledge forms, especially medical histories of query patients themselves and prototypical cases. In ISOR prototypes are used in two ways, namely in the form of guidelines for dose calculations and as generalised solutions for therapy inefficacy.

4.1 Computing initial doses: Guidelines as Prototypes

For hypothyroidism only one drug exists, namely Levothyroxine. The problem is to calculate effective initial doses. Firstly, a couple of prototypes exist. These are recommendations that have been defined by expert commissions. Though we are not sure whether they are officially accepted, we call them guidelines. The assignment of a patient to a fitting guideline is obvious because of the way the guidelines have been defined. With the help of these guidelines a range for good doses can be calculated. To compute a dose with best expected impact, we retrieve similar cases whose initial doses are within the calculated ranges. Since cases are described by few attributes and since our case base is rather small, we use Tversky's sequential measure of dissimilarity [19]. On the basis of those retrieved cases that had best therapy results an average initial therapy is calculated. Best therapy results can be determined by values of a blood test after two weeks of treatment with the initial dose. The opposite idea to consider cases with bad therapy results does not work here, because bad results can also be caused by various other reasons.

4.2 Generalised solutions for therapy inefficacy

In medical practice, therapies prescribed according to a certain diagnosis sometimes do not give desired results. Sometimes therapies are effective for some time but suddenly stop helping any more. There are many different reasons. A diagnosis might have been erroneous, the state of a patient might have changed completely or the state might have changed just slightly but with important implications for an existing therapy. Furthermore, a patient might have caught an additional disease, some other complication might have occurred, or a patient might have changed his/her lifestyle (e.g. started a diet) and so on.

For diagnosis and therapy of hypothyroidism (insufficiency of the thyroid gland) complete and elaborate knowledge exists (e.g. [27]). The diagnosis is based on analyses of laboratory results. Hypothyroidic patients are treated with hormonal therapy in the form of levothyroxine. With a proper dose of levothyroxine the states of hypothyroidic patients should become stable. If the achieved stability is disturbed, it is necessary to ascertain the reason of the disturbance and to eliminate it. Usually, a hypothyroidic patient should visit his/her doctor twice a year. Between these visits certain changes may occur in his/her condition. Some of them the doctor can find out during the standard interrogatory of the patient. These include drugs prescribed by another physician, new diseases, physiological conditions, pregnancy etc. However, there are changes that are much more difficult to find out, because a patient often does not attach significance to some changes in his/her diet and lifestyle. Therefore substantial facts may escape the attention of the doctor. The task of ISOR is to help catching these seemingly insignificant facts. Furthermore, it should indicate reasons that prevent the expected effects of the drug. ISOR first attempts to find causes for inefficacy and subsequently computes new therapy recommendations that should perform better than those administered before. Apart from a case base ISOR consists of a knowledge base, which contains therapies, conflicts, instructions and so on, of medical histories, and of prototypes. For details about the architecture of ISOR see [26]. In ISOR, prototypes are generalized cases with general solutions. They play a particular role, because they help to select a proper solution from the list of probable or available solutions. A prototype may help to point out a reason of inefficacy or it may support the doctor’s choice of a drug. Since ISOR is problem oriented, cases have appropriate attributes, namely three main ones: problem code, diagnosis and therapy. Each combination of these attribute values has its specific additional attributes. The combination A4 (problem code), hypothyroidism (diagnosis), and levothyroxine (therapy), for example, has the additional attributes age, sex, prescribed drugs, and supervisor (yes/no). Sets of cases with equal main attribute values and with similar additional attribute values build a prototype. For the combination mentioned above, one prototype P-1 is, for example, characterized by these additional values:

P-1: Age between 15 and 19, female, no prescribed drugs, and no supervisor.

The prototypes do not have unique but four general solutions:

A. irregular levothyroxine intake.
B. changes in the hormonal condition of the patient.
C. uncontrolled intake of any substance that inhibits levothyroxine absorption.
D. Intake of prescribed drugs that can inhibit levothyroxine absorption.

The additional attribute values determine the order in which these solutions are probable. For the prototype P-1 the order is B, C, and A, whereas solution D is eliminated. Some of the general solutions are specialized into some more specific solutions.

5 Summary: The Role of Prototypes

The presented systems have one thing in common that distinguishes them from most CBR systems: They use prototypes as a form of knowledge representation that fills the gap between specific cases and general rules. The main purpose of such generalised knowledge is to structure the case base, to guide the retrieval process, and sometimes to decrease the amount of storage by erasing redundant cases. In domains with very poor domain theories they may help to learn general knowledge. In domains with rather weak domain theories another advantage of case-oriented techniques is their ability to learn from cases. Only gathering new cases may improve the system‘s ability to find suitable similar cases for current problems, but it does not elicit the intrinsic knowledge of the stored cases. To learn the knowledge contained in cases a generalisation process is necessary. In our early warning system concerning the kidney function, apart from guiding the retrieval and structuring the case base prototypes mainly serve to learn typical course patterns, because just the relevant kidney parameters are known but no knowledge about their temporal course behaviour exists.

For diagnosis of dysmophic syndromes prototypes correspond directly to the physician‘s sense of prototypes. As comparisons with single cases are unable to identify typical features, in this application the use of prototypes is not only sensible, but even necessary.

In ISOR, the prototypes for dose calculation are guidelines and the prototypes for therapy inefficacy are similar to those for diagnosis of dysmorphic syndromes. The main difference is that in ISOR all prototypes are defined by medical experts.

Summarising our experiences we would like to make quite clear that the role of prototypes depends on the application and the task. For medical diagnoses they even seem to be necessary because of their correspondence to medical prototypes which guide the physicians diagnoses.

References

1. Strube, G., Janetzko, D.: Episodisches Wissen und fallbasiertes Schließen: Aufgaben für die Wissensdiagnostik und die Wissenspsychologie. Schweizerische Zeitschrift für Psychologie 49 (4), 211-221 (1990)
2. Swanson, D.B., Feltovich, P.J., Johnson, P.E.: Psychological Analysis of Physician Expertise: Implications for Design of Decision Support Systems. In: Shires, D.B., Wolf, H. (eds.) MEDINFO 77 , pp. 161-164 North-Holland, Amsterdam (1977)
3. Schank, R.C.: Dynamic Memory: A Theory of Learning in Computer and People. Cambridge University Press, New York (1982)
4. Bareiss, R.: Exemplar-based Knowledge Acquisition. Academic Press, San Diego (1989)
5. Maximini, K., Maximini, R., Bergmann R.: An Investigation of Generalized Cases. In: Asley, K.D., Bridge, D.G. (eds.) ICCBR 2003, pp. 261-275 Springer, Heidelberg, 261-275 (2003)
6. Bellazzi, R., Montani, S., Portinale, L.: Retrieval in a Prototype-based Case Library: a Case Study in Diabetes Therapy Revision. In: Smyth, B., Cunningham, P. (eds) EWCBR, pp. 64-75,Springer, Heidelberg (1998)
7. Bichindaritz, I.: Case-based Reasoning Adaptive to Several Cognitive Tasks. In: Aamodt, A., Veloso, M. (eds) ICCBR-95, pp. 391-400, Springer, Heidelberg (1995)
8. Turner, R.: Organizing and Using Schematic Knowledge for Medical Diagnosis. In: Kolodner, J. (ed) Workshop on CBR, pp.435-446, Florida (1988)
9. Perner, P.: A Comparative Study of Catalogue-Based Classification. In: Roth-Berghofer TR et al (eds.) ECCBR, pp. 301-308, Springer, Heidelberg, (2006)
10. Bichindaritz, I.: Prototypical Cases for Knowledge Maintenance for Biomedical CBR. In: Weber, R, Richter, MM. (eds.) ICCBR 2007, pp. 492-506, Springer, Heidelberg (2007)
11. Clancey, W.J.:Heuristic Classification. Artificial Intelligence 27, 289-350 (1985)
12. Selz, O.: Über die Gesetze des geordneten Denkverlaufs. Stuttgart (1913)
13. Perner, P.: Are Case-based Reasoning and Dissimilarity-based Classification Two Sides of the Same Coin? Journal Engineering Applications Artif Intel, 15 (2), 205-216 (2004)
14. Schmidt, R., Waligora, T., Vorobieva, O.: Prototypes and Case-Based Reasoning for Medical Applications. In: Perner, P. (ed.) Case-Based Reasoning on Images and Signals, pp. 285-317, Springer, Heidelberg, (2008)

15. Taybi, H., Lachman, R.S.: Radiology of Syndromes, Metabolic Disorders, and Skeletal Dysplasia.. Year Book Medical Publishers, Chicago (1990)
16. Gierl, L., Stengel-Rutkowski, S.: Integrating Consultation and Semi-automatic Knowledge Acquisition in a Prototype-based Architecture: Experiences with Dysmorphic Syndromes. Artificial Intelligence in Medicine 6, 29-49 (1994)
17. Clinical Dysmorphology. www.clindysmorphol.com
18. Winter, R.M., Baraitser, M., Douglas, J.M.: A Computerised Data Base for the Diagnosis of Rare Dysmorphic Syndromes. Journal medical genetics 21 (2), 121-123 (1984)
19. Tversky, A.: Features of Similarity. Psychological Review 84 (4), 327-352 (1977)
20. Rosch, E., Mervis, C.B.: Family Resemblance: Studies in the Internal Structures of Categories. Cognitive Psychology 7, 573-605 (1975)
21. Wenkebach, U., Pollwein, B., Finsterer, U.: Visualization of Large Datasets in Intensive Care. Proc Annu Symp Comput Appl Med Care, 18-22 (1992)
22. Schmidt, R., Pollwein, B., Gierl, L.: Medical Multiparametric Time Course Prognoses Applied to Kidney Function Assessments. Int J Med Inform 53 (2-3), 253-264 (1999)
23. Shahar, Y. (1999): Timing is Everything: Temporal Reasoning and Temporal Data Maintenance in Medicine. In: Horn, W., Shahar. Y., Lindberg, G., Andreassen, S., Wyatt, J. (eds) AIMDM'99, pp. 30-46, Springer, Heidelberg, 30-46 (1999)
24. Haimowitz, I.J., Kohane, I.S.: Automated Trend Detection with Alternate Temporal Hypotheses. In: IJCAI, pp. 146-151, Morgan Kaufmann Publishers, San Mateo (1993)
25. DeSarbo, W.S. et al.: TSCALE: A new Multidemensional Scaling Procedure based on Tversky's Contrast Model. Psychometrika 57, 43-69 (1992)
26. Schmidt, R., Vorobieva, O.: Case-Based Reasoning Investigation of Therapy Inefficacy. Knowledge-Based Systems 19 (5), 333-340 (2006)
27. DeGroot, L.J.: Thyroid Physiology and Hypothyroidsm. In: Besser, G.M., Turner, M. (eds.) Clinical endocrinology. Wolfe, London (Chapter 15) (1994)