Case Based Representation and Retrieval with Time Dependent Features

Stefania Montani and Luigi Portinale

Dipartimento di Informatica, Università del Piemonte Orientale,
Alessandria, Italy

Abstract. The temporal dimension of the knowledge embedded in cases has often been neglected or oversimplified in Case Based Reasoning systems. However, in several real world problems a case should capture the evolution of the observed phenomenon over time. To this end, we propose to represent temporal information at two levels: (1) at the *case level*, if some features describe parameters varying within a period of time (which corresponds to the case duration), and are therefore collected in the form of time series; (2) at the *history level*, if the evolution of the system can be reconstructed by retrieving temporally related cases.

In this paper, we describe a framework for case representation and retrieval able to take into account the temporal dimension, and meant to be used in any time dependent domain. In particular, to support case retrieval, we provide an analysis of similarity-based time series retrieval techniques; to support history retrieval, we introduce possible ways to summarize the case content, together with the corresponding strategies for identifying similar instances in the knowledge base. A concrete application of our framework is represented by the system RHENE, which is briefly sketched here, and extensively described in [20].

1 Introduction

The Case Based Reasoning (CBR) methodology [1] is particularly appealing in those domains where acquiring and formalizing knowledge would be a significantly hard and time consuming task.

As a matter of fact, CBR allows one to build a knowledge base of past situations (*cases*), which represent an operative form of knowledge, that can be reused in present problems, possibly after an adaptation step. Representing a real-world situation as a case is often straighforward: given a set of meaningful features for the application domain, it is sufficient to identify the value they assume in the situation at hand; sometimes a case also stores information about the solution applied and the outcome obtained. Due to this quick procedure, in many applications the knowledge acquisition bottleneck can be extremely reduced with respect to the exploitation of other reasoning methodologies.

The relative simplicity of defining cases has often led researchers to neglect or oversimplify a very important aspect of the knowledge embedded in past situations: the *temporal dimension*. On the other hand, in several (especially medical)

H. Muñoz-Avila and F. Ricci (Eds.): ICCBR 2005, LNCS 3620, pp. 353–367, 2005.

applications, the need of accounting for time is widely recognized. Actually, in many domains cases cannot be interpreted merely as snapshots of the world at a given time instant: in a lot of real problems a case should capture the evolution of the observed phenomenon over time. In medical practice, for example, before prescribing a therapy (i.e. the case solution) the physician needs to keep in mind the clinical history that led the patient to the current situation; actually, the pattern of the patient's changes is often more important than the final state [18]. Similarly, forecasting tasks often require an analysis of temporal sequences of observations or of interactions between involved agents [25]. The definition of a case as a set of feature/value pairs needs therefore to be refined.

In particular, we envision the possibility of addressing the temporal dimension at two levels:

1. at the *case level*, if some features describe parameters varying within a period of time (which corresponds to the case duration), and are therefore collected in the form of time series;
2. at the *history level*, if the evolution of the system can be reconstructed by retrieving temporally related cases (e.g. in a medical domain, cases concerning consecutive visits of a given patient).

As an example, in hemodialysis treatment it is possible to define a case as a dialysis session, which includes time series features, that justify the need of accounting for temporal information at the case level. Moreover, in clinical practice physicians use to judge the patient's behaviour in the latest two weeks (i.e. they deal with a history of a few consecutive cases); only in particularly critical situations, they enter the detail of single sessions. Both levels are therefore needed in this context.

If we want to guarantee consistent results, we have to take into account the fact that the temporal dimension complicates not only the knowledge representation task, but the retrieval process as well. In particular, similarity-based time series retrieval has to be addressed on the one hand, while strategies for matching patterns made by "consecutive" cases needs to be defined.

In this paper, we describe a framework for case-based representation and retrieval meant to be used in any time dependent domain. In particular, in section 2 we describe related works; in section 3, we deal with knowledge representation and retrieval at the *case level*, while in section 4 we extend our discussion to the *history level*. Section 5 describes how our theoretical work is being applied in the system RHENE [20], a tool for managing patients in a hemodyalisis regimen. Finally, section 6 is devoted to conclusions and future work.

2 Related Work

Despite the need of accounting for the temporal dimension in a CBR system may appear important, rather interestingly the representation of time-dependent information and its impact on the CBR cycle [1] have been scarcely inspected in the literature.

Actually, just a few works in this sense exist. Most of them afford the problem of representing and retrieving cases with time-extended features (i.e. time series), and each work is substantially limited to fit a single application domain: robot control [24], process forecast [21,26], process supervision [10], prediction of faulty situations [14] and time course prognoses for medical problems [27]. Almost all of these contributions adopt a representation of temporal knowledge requiring that absolute time *points* are associated with the temporal objects being modelled. This hypothesis may be unrealistic in many applications, where only relative, and often qualitative, temporal knowledge is available; a more suitable *interval*-based model [3] has been chosen only in [14] (see section 3.1 for details on these knowledge representation concepts). Moreover, all works share two main limitations: (1) in most cases, since they have been thought to support a specific application, their generalizability is limited or not discussed at all; (2) they address the temporal dimension only at the *case level*. With respect to issue 1, actually a more general framework for case representation and retrieval with time-dependent features has been proposed in [13]; this paper deals with the problem of time series similarity and proposes a complex retrieval strategy; nevertheless, it is still limited to the *case level* temporal dimension.

On the other hand, a recent contribution [29] deals with temporal information at the *history level*, in the respiratory sinus arrhythmia domain. More interestingly, [19] presents an application independent logic formalism addressing history representation. From the temporal model point of view, this work is particularly interesting because it accommodates both points and intervals as primitive time elements. Nevertheless, how to deal with retrieval is not described; the authors only claim that graph similarity algorithms could be adopted. Moreover, they still do not address the temporal dimension in CBR as a whole, because features in the form of time series are not taken into account.

Finally, temporal knowledge representation for CBR is discussed in [8]. In this work, both points and intervals are exploited as well. However, here a clear distinction between cases and histories is not provided. In particular, a single case captures the overall evolution of the system under observation (i.e. the patient, since the work is applied to a medical domain), but snapshots of the feature values, limited to specific time intervals, are used for retrieval. Thus, to our knowledge, our work represent a meaningful effort towards a more comprehensive treatment of the two levels of the temporal dimension, as introduced in section 1.

3 The Temporal Dimension in Case-Based Retrieval: The Case Level

3.1 Case Representation

In our framework, we adopt a model for representing temporal information based on both the *point* and the *interval* primitives, in order to deal with as much real world situations as possible (see also [19]). In particular:

- a *point* is identified by an absolute (i.e. numeric) or relative (i.e. qualitative) temporal coordinate, expressed with respect to the reference system and the granularity of the application domain;
- an *interval* is identified by an ordered pair of points, which represent its starting point and its ending point respectively.

Given these premises, we have to detail what we mean for *case*, and how temporal information at the case level can be formalized.

As previously observed, in some real world applications, it may be limiting to conceive a case as an instantaneous situation, where all feature values are singletons and remain unchanged. In our framework, therefore, some features can take the form of (typically discretized) uni-dimensional time series. In addition, we associate to each case an *interval* - Case Interval (CI) henceforth - meant to represent the period of time in which all the feature values were measured.

With respect to the CI, features must satisfy these requirements:

1. each feature in the form of a single value has to be measured at a time point which is included between the starting and the ending point of the CI;
2. for a feature in the form of a time series, each value has to respect requirement (1) above.

3.2 Case Retrieval

Case retrieval needs to cope with the different types of features that can be defined in a case: time stamped single valued data points, and time series (we make the hypothesis that all features values at the case level are raw data).

Although it is not necessary, the different nature of features could suggest to treat them in different ways in the retrieval process. Without the expectation of providing an exhaustive panorama of alternatives, we would like to propose a modular architecture, that appears relatively general, in the sense that its elements can be skipped or differently combined, in order to obtain new solutions.

The proposed retrieval process (see figure 1) can be sketched as follows (at a very high level):

- use (some) single valued features for a *classification/grouping step*, to reduce the search space for retrieval itself;
- perform a *multi-step retrieval* in the output class:
  1. select some particularly relevant time series features;
  2. search for a set of cases similar to the query one in the direction of one of the selected features at a time;
  3. provide some kind of *combination* of the sets of locally similar cases identified above;
  4. order the results on the basis of all features, including also time stamped data points.

As regards the *combination* of locally similar cases, to be merged into a unique set, different alternatives may be devised. A possible combination function is

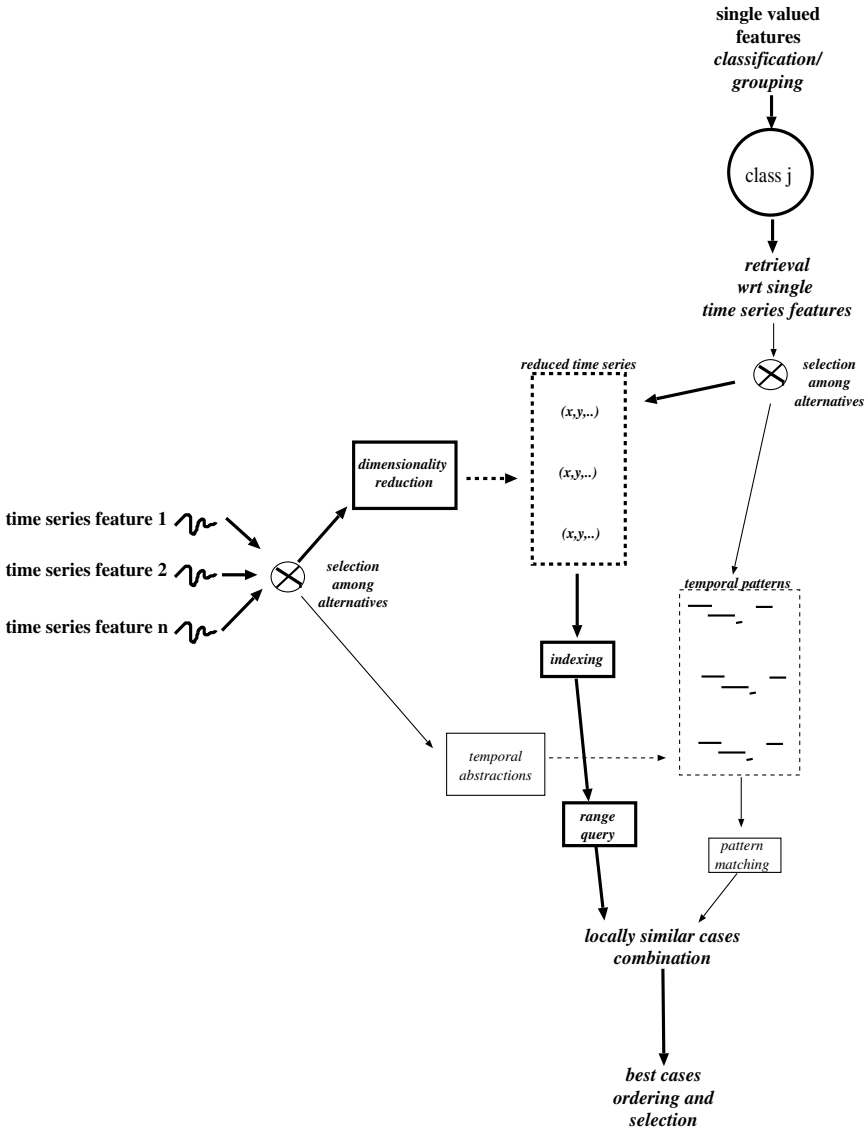

Fig. 1. A general architecture for case retrieval with time varying features. A classification/grouping step may be used to reduce the search space. Retrieval then takes place, in the direction of a single time series feature at a time. To optimize similarity-based time series retrieval, it is possible to select one out of two alternatives: (1) reduce dimensionality (e.g. by applying DFT) and then exploit spatial indexing techniques; (2) summarize the raw data by applying Temporal Abstractions (TA) and then exploit pattern matching techniques. Locally similar cases are then properly combined to produce the final output. The modules of the general architecture that have been implemented in the system RHENE (namely: classification, dimensionality reduction, indexing, range query and combination - see section 5) are highlighted in bold. Notice that, even if not shown in the figure, the process involves a query represented by raw time series data that are reduced or abstracted depending on the retrieval technique that is used (i.e. range query on an index structure or pattern matching over TA)

intersection. Suppose that locally similar cases were extracted through a set of range queries, one in each feature's direction; intersection extracts the cases that satisfy the request of being within all the specified ranges of similarity contemporaneously. Clearly this is quite a strong requirement. A less strict result may be obtained by using union as a combination function. In this hypothesis, a case will be globally accepted if it belongs at least to one range of similarity. Clearly, other combination operators may be introduced as well.

As a concrete example of multistep retrieval, section 5 describes the architecture of the system RHENE [20], developed by the authors in collaboration with the University of Pavia in Italy. RHENE's architecture instantiates a subset of the modules of figure 1, highlighted in bold.

While the retrieval of cases with single valued features is a classical topic of CBR, we can spend a few words on the retrieval of cases with time series features. For the sake of clarity, we will concentrate on the search of cases similar to the input one in the direction of one particular parameter, which is in the form of a discretized time series.

A wide literature exists about how to optimize similarity-based retrieval of time series. Before entering the details, we propose to distinguish between two main directions:

- apply a dimensionality reduction technique;
- summarize the raw data by means of a technique able to derive higher level information from them, such as Temporal Abstractions [6].

Blocks applying these methods in the general retrieval architecture can be recognized in figure 1. The following subsections provide a deeper insight of these two alternative procedures.

Dimensionality Reduction. In the literature, most of the approaches to similarity-based time series retrieval are founded on the common premise of dimensionality reduction (see the survey in [12]).

As a matter of fact, a discretized time series can always be seen as vector in an n-dimensional space (with n typically extremely large). Simple algorithms for retrieving similar time series take polynomial time in n. Multidimensional spatial indexing (e.g. resorting to R-trees [11]) can even lead to sub-linear retrieval; nevertheless, these tree structures are not adequate for indexing high-dimensional data sets [7].

One obvious solution is thus to reduce the time series dimensionality, by means of a transform that preserves the distance between two time series, or underestimates it: in this case a post-processing step will be required, to filter out the so-called "false alarms"; the requirement is never to overestimate the distance, so that no "false dismissals" can exist [12]. Widely used transforms are the Discrete Fourier Transform (DFT) [2], and the Discrete Wavelet Transform (DWT) [9].

DFT maps time series to the frequency domain. DFT application for dimensionality reduction stems from the observation that, for the majority of real-world

time series, the first (1-3) Fourier coefficients carry the most meaningful information, and the remaining ones can be safely discarded. Moreover, Parseval's theorem [22] guarantees that the distance in the frequency domain is the same as in the time domain, when resorting to any similarity measure that can be expressed as the Euclidean distance between feature vectors in the feature space. In particular, resorting only to the first Fourier coefficients can underestimate the real distance, but never overestimates it.

On the other hand, wavelets are basis functions used to represent other functions. The wavelet transform can be repeatedly applied to the data, obtaining that each application brings out a higher resolution of the data, while at the same time it smoothes the remaining data. The output of the DWT consists of the remaining smooth components and of all the accumulated detail components. DWT, like any orthonormal transform, preserves the Euclidean distance as the DFT does. The number of wavelet coefficients to be kept, although lower than the original data dimensionality, is often higher than in the case of DFT application.

Retrieval of series transformed either by DFT or by wavelets can then benefit from the use of spatial index structures, such as the R-tree [11], the X-tree [7], and the TV-tree [31], whose features are widely discussed in the database literature, or from other specific indexing techniques (see e.g. [23]).

A different approach to dimensionality reduction is Piecewise Constant Approximation (PCA) (see e.g. [16,17]): it consists in dividing a time series into k segments, and in using their average values as a k-dimensional feature vector (where obviously $k << n$, the original data dimensionality). The best value of k can also be estimated.

The choice of the most cost-effective transformation to apply should be done on the basis of the application at hand.

Temporal Abstractions. While dimensionality reduction is a widely accepted technique for optimizing similarity-based retrieval of time series, the use of Temporal Abstractions (TA) [28,6] in this field is not often reported. Nevertheless, we believe it represents a valuable alternative to dimensionality reduction itself, in particular when:

- a more *qualitative* abstraction of the time series values is sufficient;
- a clear mapping between raw and transformed data has to be made available;
- the mapping itself needs to be easily interpretable by end users as well.

TA is an Artificial Intelligence methodology able to solve a *data interpretation task* [28], whose goal is the one of deriving high level concepts from time stamped data. Through TA, huge amounts of temporal information, like the one embedded in a time series, can be effectively mapped to a compact representation, that not only summarizes the original longitudinal data, but also abstracts meaningful behaviours in the data themselves.

Operatively, the basic principle of TA methods is to move from a *point-based* to an *interval-based* representation of the data [6], where: (i) the input

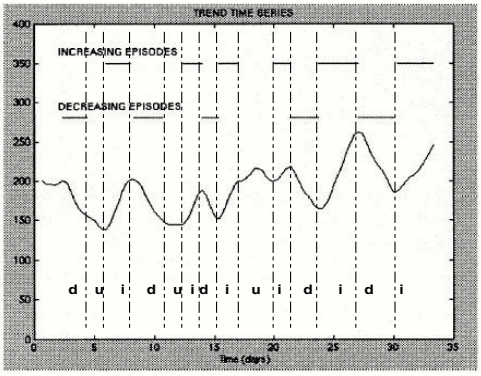

Fig. 2. An example of trend TA, applied to a blood glucose level time series [5]. The abstraction produces a pattern where symbols d, i, u stand for *decreasing*, *increasing* and *undecided* respectively

points (*events* henceforth) are the elements of the discretized time series; (ii) the output intervals (*episodes* henceforth) aggregate adjacent events sharing a common behaviour, persistent over time. More precisely, the method described above should be referred to as *basic* TA [6].

Basic abstractions can be further subdivided into *state* TA and *trend* TA. *State* TA are used to extract episodes associated to *qualitative levels* of the monitored feature, e.g. low, normal, high values; *trend* TA are exploited to detect specific *patterns*, such as increase, decrease or stationarity, from the time series. The output results of a basic TA depend on the value assigned to specific parameters, such as the granularity (the maximum temporal gap between two events allowed for aggregating them into the same episode) and the minimum extent (the minimum time extent for considering an episode relevant) for state TA, and the slope (the minimum allowed rate of change in an episode) for trend TA.

Complex TA [6] can be defined as well: instead of aggregating events into episodes, complex TA aggregate two series of episodes into a set of episodes of higher level (i.e., they abstract output intervals over precalculated input intervals). In particular, complex abstractions search for specific *temporal* relationships between episodes which can be generated from a basic abstraction or from other complex abstractions. The relation between intervals can be any of the temporal relations defined by Allen [3]. This kind of TA can be exploited to extract patterns that depend on the course of several features, or to detect patterns of complex shapes in a single feature.

If the time series has been pre-processed through TA, similarity based retrieval can benefit of the use of pattern matching techniques. Sequence matching can in fact be performed by a number of well-established methods [30] like dynamic programming based on edit distance approach [32], suffix tree-based approaches [33] or general formal transformations of patterns [15]. For example the framework in [15] defines similarity between a pattern A and a pattern B

(in a formal pattern language P) as a function of the transformations (defined on a transformation language T) needed to reduce B to A (or vice versa). The approach allows one to answer also queries such as "find all patterns similar to some pattern A, but not similar to pattern B". Figure 2 shows an example of a trend TA producing a pattern (over a granularity based on days) where symbols d, i, u stand for *decreasing*, *increasing* and *undecided* respectively.

Finally, we can notice that the use of TA can be limited to query the case library, if we do not want to explicitly abstract raw time series data, but we still want to maintain the capability of using the language of TA at the query level. For example, [34] introduces an algorithm where a symbolic query (in the form of sequence of symbols like those produced by a TA) can be answered over a database of raw time series data, by producing those subsequences that best match the query itself, following specific abstraction rules (like for instance those that may be used to define a TA).

4 The Temporal Dimension in Case-Based Retrieval: The History Level

By history we mean a set of temporally related or time consecutive cases, which refer to the same "object" or "entity" (e.g. the same patient in a medical domain, or the same class of devices in a fault diagnosis domain). Histories could be of various length; actually the number of cases that compose a history is a typical application dependent parameter. Histories themselves could be built "on the fly", when instances similar to the input one have to be retrieved; alternatively, they may be precompiled, and stored in a memory in which history search will then take place. In the case of precompilation, supposing that the history length is a known parameter[1], all possible histories could be built from the library of cases, or just a subset of them. The system could then precalculate all the histories for the given patient (or for all the patients in the case base), within the given time window. Of course, a trade-off exists between the cost of precompilation (and history storage) and the complexity of retrieval if histories have to be built just at retrieval time.

History retrieval can be the only goal of the retrieval system or it can be exploited as a search space reduction step (alternative to classification/grouping, see section 3.2), to be followed by case retrieval itself, which will then be focused only on the cases composing the retrieved histories. Figure 3 presents an architecture where history retrieval provides the first results, at a high level of abstraction; if the user is interested in more details, case retrieval (on a search

[1] This is not necessarily an unrealistic assumption. Actually, in some (e.g. medical) applications, the temporal window to take into account (i.e. the history length) could be well identified on the basis of the domain knowledge, and could also be explicitly provided by a guideline. For example, in hemodialysis treatment, the temporal window is normally made of two weeks, which correspond to a sequence of 6 consecutive cases.

space shrinked as described above), will refine the output, concentrating e.g. on more specific features values (at the case level).

4.1 History Representation

It is worth noting that history retrieval requires less detailed information with respect to case retrieval: the modelled object behaviour is being observed from a higher level perspective; therefore, for each case composing the history, the case content has to be somehow summarized.

To this hand, we envision different possibilities:

- first, a single value, "valid" in all the CI, can be assigned to each feature (see also [19]). This is trivially the case if the feature is a single data point. Dealing with time series, on the other hand, the value could correspond e.g. to the mean, or to the most frequent value in the feature measurements; more interestingly, it could be obtained as a *pattern approximately stable over the CI*, typically extracted through TA techniques, applied to the original data;
- as a second possibility, summarization can be obtained through a *granularity change*: a history can be interpreted as a "macro-case", whose features derive from the corresponding features in the cases composing the history. For time series, the macro-case features would be time series (inter-case data) of multidimensional time series (intra-case data). This information needs to be synthesized, for example through some sufficient statistics indexes, such as the median and the 10th and 90th percentiles of each variable. The macro-case features would then become the series of the medians and of the percentiles (or simply the series of the values for single valued features) over all the cases reported in the history. In the resulting macro-case, all features will then be in the form of time series. This second possibility seems more easily applicable if cases don't overlap in time.

In the next section, we discuss proper retrieval strategies for both the summarization alternatives.

4.2 History Retrieval

When each case has been mapped to a *pattern stable over an interval*, TA and pattern matching techniques immediately appear as good candidates for retrieval.

In particular, when intervals are the input to the TA process, complex TA (see section 3.2) can be applied to extract temporal patterns in the history, that correspond to significant behaviours in the process being observed. For example, a peak in a case feature f defined by two consecutive cases can be identified by a complex TA of the form "an increasing trend in f *meets* a decreasing trend in f", where *meets* is an operator of Allen's interval algebra [3]. The mechanism can still be applied if the cases are (partially) overlapping. Once meaningful patterns have been identified in the query history, similar histories can be extracted relying upon pattern matching techniques. As for the case level,

various retrieval architectures could be designed; typically, some features could be more relevant than others in history retrieval, and could be used for the selection of very relevant histories, to be then ordered on the basis of all feature values (see section 3.2 and figure 3).

On the other hand, if a *granularity change* has been applied, the problem is basically reduced to case retrieval, and the considerations of section 3.2 hold. In this situation, all case features are in the form of time series, that require a preprocessing for optimizing retrieval itself. Since in history retrieval the goal is the one of abstracting higher level concepts from raw data, the use of TA appears particularly appealing in this case as well. Pattern matching techniques will then help for a similarity-based search in the case memory (see figure 3). In particular, some ground cases features are now mapped to more than one history features (e.g. median and percentiles, or mean and standard deviation). The different meaning of these features could correspond to a different role in the retrieval process. For example, a preprocessing step could filter out histories in which standard deviation values are too high. Alternatively, if a weight defines the importance of each feature (at the case level), the weight of a case feature that has been mapped to a mean and a standard deviation at the history level could be decomposed in two numbers, to be assigned as the weights of the mean and of the standard deviation respectively. A combination (e.g. the product) of the two numbers (at the history level) would provide the weight of the original feature at the case level.

5 The Framework in Practice: The RHENE System

RHENE (**R**etrieval of **HE**modialysis in **NE**phrological disorders) is a multi-step case retrieval system applied to the domain of patients affected by nephropatologies and treated with hemodialysis [20]. Defining a dialysis session as a case, retrieval (at the case level) has to operate both on single valued and time series features.

RHENE implements a subpart of the modules of the general architecture in figure 1 (highlighted in bold).

In particular, a preliminary classification/grouping step, based on single-valued features, reduces the retrieval search space. Intra-class retrieval then takes place by considering time series features, and is articulated as follows: (1) locally similar cases (considering one feature at a time) are extracted and the *intersection* of the retrieved sets is computed; (2) global similarity is computed, as a *weighted average* of local distances, and the best cases are listed. For similarity-based time series retrieval (step (1)), we rely on dimensionality reduction, and in particular on DFT. Thanks to specific index structures (i.e. k-d trees and TV trees) range queries can be efficiently performed on our case base. Both ranges and weights are tunable parameters; this choice provides the tool with great flexibility.

The current prototype has been positively tested on a case base of more than 6500 cases, belonging to 48 real patients.

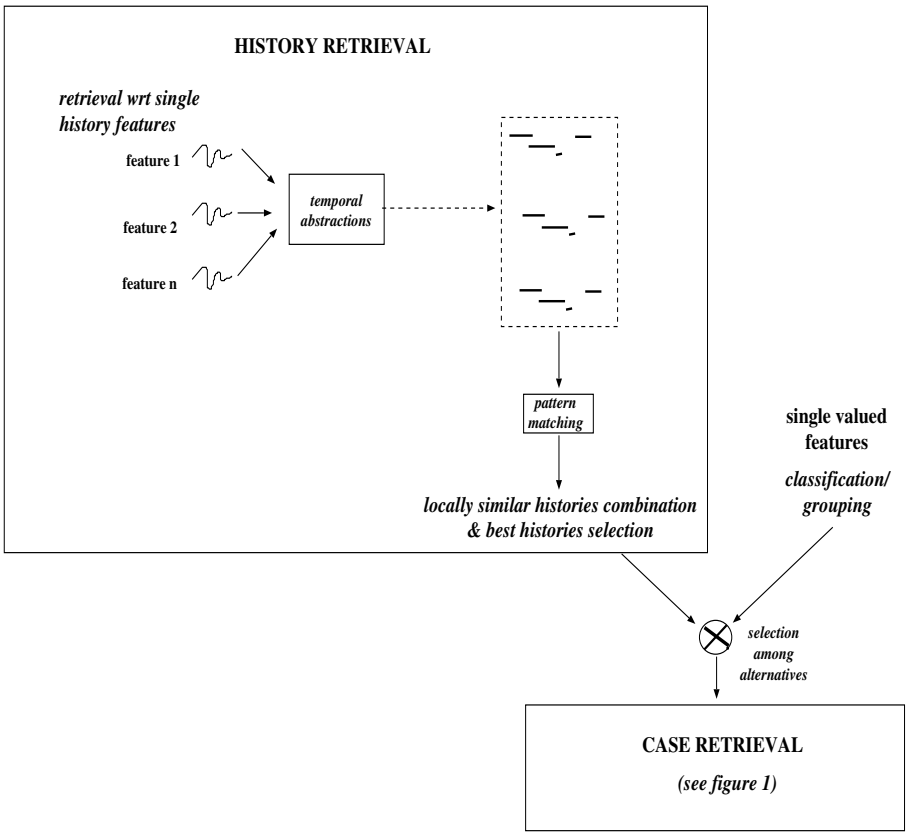

Fig. 3. A general retrieval architecture, in which history retrieval can be used, in alternative to classification/grouping, to reduce the search space for case retrieval. Case retrieval can then be exploited to obtain more detailed results, concentrating on more specific features values. History retrieval is sped up by the use of TA and of pattern matching techniques. For the case retrieval block, please refer to figure 1. As in case of figure 1, we omit here to explicitly show the query

In the future, we plan to work at the history level as well, by redefining a case as a longer monitoring period (see section 4), typically made by all the dialysis sessions of a patient within two weeks. As a matter of fact, as observed in the introduction, this enlarged granularity is closer to the viewpont from which physicians use to evaluate the dialysis data and to judge the patient's evolution over time. A tool (called EMOSTAT) able to summarize the raw data along these lines, and to provide an off-line monitoring facility to nephrologists, has already been implemented at the University of Pavia [4]. In particular, in EMOSTAT time series data are synthesized through the median and the 10th and 90th percentiles of each monitoring variable.

This tool is going to be integrated with RHENE[2], in order to implement history retrieval. On the history features, we want to look for particular patterns (e.g. episodes of increasing values, peaks, etc.), that we will highlight by preprocessing the data through TA, and by applying approximate string matching techniques. As a second step, the physician will be allowed to enter the detail of the cases composing the retrieved histories, and formulate stricter queries, on the basis of feature values of particular interest. History retrieval will therefore be available as an autonomous facility, or as a preprocessing step for case retrieval, as described in figure 3. The overall architecture resulting from the integration of the two systems will provide a support for patient examination and therapy evaluation, but could also be adopted as a means for assessing the quality of the hemodialysis service, producing a useful input from the knowledge management perspective. Technically speaking, quality assessment requires to fulfil two tasks: (1) discover relationships between the time patterns of the process data and the performance outcomes; (2) retrieve similar critical patterns within the process data, in order to assess their frequency. While EMOSTAT is able to address task (1), the role of RHENE is the one of implementing task (2), thus providing a comprehensive approach towards the realization of an auditing procedure, able to summarize the dialysis sessions from a clinical quality viewpoint.

6 Conclusions and Future Works

In this paper, we have presented a domain-independent framework for dealing with the temporal dimension in case representation and retrieval. In particular, we have proposed a multi-step retrieval architecture whose modules can be differently instantiated and combined, in order to cover the various needs of the possible application domains. An example of implementation is represented by the system RHENE, described in section 5. RHENE currently implements a subpart of the overall architecture, limited to the case level. In the future, we plan to deal with knowledge representation and retrieval at the history level as well, in order to provide physicians with a more flexible tool, that will enable them to inspect patients' data by referring to different time granularities. This work, which will be supported by a grant of the Italian Ministry of Education, will be limited to a specific application domain. Nevertheless, it will represent a first step towards a better understanding of the advantages possibly provided by the methodology proposed in this paper, and will allow us to inspect its usability in practice.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and systems approaches. *AI Communications*, 7:39–59, 1994.

[2] This work will be supported by the grant PRIN 2004 number 2004094558, funded by the Italian Ministry of Education.

2. R. Agrawal, C. Faloutsos, and A.N. Swami. Efficient similarity search in sequence databases. In D. Lomet, editor, *Proc. 4th Int. Conf. of Foundations of Data Organization and Algorithms*, pages 69–84. Springer-Verlag, Berlin, 1993.
3. J.F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.
4. R. Bellazzi, C. Larizza, P. Magni, and R. Bellazzi. Temporal data mining for the quality assessment of a hemodialysis service. *Artificial Intelligence in Medicine (in press)*.
5. R. Bellazzi, C. Larizza, P. Magni, S. Montani, and M. Stefanelli. Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. *Artificial Intelligence in Medicine*, 20:37–57, 2000.
6. R. Bellazzi, C. Larizza, and A. Riva. Temporal abstractions for interpreting diabetic patients monitoring data. *Intelligent Data Analysis*, 2:97–122, 1998.
7. S. Berchtold, D.A. Keim, and H.P. Kriegel. The x-tree: an index structure for high-dimensional data. In *Proc. VLDB 96*, pages 28–39. Morgan Kaufman, San Mateo, CA, 1996.
8. I. Bichindaritz and E. Conlon. Temporal knowledge representation and organization for case-based reasoning. In *Proc. TIME-96*, pages 152–159. IEEE Computer Society Press, Washington, DC, 1996.
9. K.P. Chan and A.W.C. Fu. Efficient time series matching by wavelets. In *Proc. ICDE 99*, pages 126–133. IEEE Computer Society Press, Washington, DC, 1999.
10. B. Fuch, A. Mille, and B. Chiron. Operator decision aiding by adaptation of supervision strategies. In *Case-Based Reasoning Research and Development, LNAI*, pages 23–32. Springer-Verlag, Berlin, 1995.
11. A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. ACM SIGMOD*, pages 47–57. ACM Press, New York, NY, 1984.
12. M.L. Hetland. A survey of recent methods for efficient retrieval of similar time sequences. In H. Bunke M. Last, A. Kandel, editor, *Data Mining in Time Series Databases*. World Scientific, London, 2003.
13. M. Jaczynski. A framework for the management of past experiences with time-extended situations. In *Proc. ACM conference on Information and Knowledge Management (CIKM) 1997*, pages 32–38. ACM Press, New York, NY, 1997.
14. M.D. Jaere, A. Aamodt, and P. Skalle. Representing temporal knowledge for case-based prediction. In S. Craw and A. Preece, editors, *Proc. European Conference on Case Based Reasoning (ECCBR) 2002, in: Lecture Notes in Artificial Intelligence 2416*, pages 174–188. Springer-Verlag, Berlin, 2002.
15. H.V. Jagadish, A.O. Mendelzon, and T. Milo. Similarity based queries. In *Proc. 14th ACM Symp. on Principles of Database Systems*, San Jose, CA, 1995.
16. E. Keogh. Fast similarity search in the presence of longitudinal scaling in time series databases. In *Proc. Int. Conf. on Tools with Artificial Intelligence*, pages 578–584. IEEE Computer Society Press, Washington, DC, 1997.
17. E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, 2000.
18. E.T. Keravnou. Modeling medical concepts as time objects. In *Proceedings AIME 1995, LNAI 934*, pages 67–90. Springer-Verlag, Berlin, 1995.
19. J. Ma and B. Knight. A framework for historical case-based reasoning. In K.D. Ashley and D.G. Bridge, editors, *Proc. International Conference on Case Based Reasoning (ICCBR) 2003, in: Lecture Notes in Artificial Intelligence 2689*, pages 246–260. Springer-Verlag, Berlin, 2003.

20. S. Montani, L. Portinale, R. Bellazzi, and G. Leonardi. Rhene: a case retrieval system for hemodialysis cases with dynamically monitored parameters. In P. Funk and P.A. Gonzales Calero, editors, *Proc. European Conference on Case Based Reasoning (ECCBR) 2004, in: Lecture Notes in Artificial Intelligence 3155*, pages 659–672. Springer-Verlag Berlin, 2004.
21. G. Nakhaeizadeh. Learning prediction from time series: a theoretical and empirical comparison of cbr with some other approaches. In *Topics in Case-Based Reasoning, LNAI 837*, pages 65–76. Springer-Verlag, Berlin, 1994.
22. A.V. Oppenheim and R.W. Shafer. *Digital signal processing*. Prentice Hall, London, 1975.
23. D. Patterson, M. Galushka, and N. Rooney. An effective indexing and retrieval approach for temporal cases. In *Proc. 17th FLAIRS 2004, AAAI Press, Miami*, 2004.
24. A. Ram and J.C. Santamaria. Continuous case-based reasoning. In *Proc. AAAI Case-Based Reasoning Workshop*, pages 86–93, 1993.
25. F.E. Ritter and J.H. Larkin. Developing process models as summaris of hci action sequences. *Human Computer Interaction*, 9:345–383, 1994.
26. S. Rougegrez. Similarity evaluation between observed behaviours for the prediction of processes. In *Topics in Case-Based Reasoning, LNAI 837*, pages 155–166. Springer-Verlag, Berlin, 1994.
27. R. Schmidt, B. Heindl, B. Pollwein, and L. Gierl. Abstraction of data and time for multiparametric time course prognoses. In *Advances of Case-Based Reasoning, LNAI 1168*, pages 377–391. Springer-Verlag, Berlin, 1996.
28. Y. Shahar. A framework for knowledge-based temporal abstractions. *Artificial Intelligence*, 90:79–133, 1997.
29. M. Sollenborn and M. Nilsson. Building a case-base for stress diagnosis: an analysis of classified respiratory sinus arrhythmia sequences. In *Proc. Case-Based Reasoning in the Health Sciences Workshop, European Conference on Case Based Reasoning (ECCBR) 2004*.
30. G.A. Stephen. String searching algorithms. In *Lecture Notes Series in Computing*, volume 3. World Scientific, 1994.
31. V.S. Subrahmanian. *Principles of Multimedia Database Systems*. Morgan Kaufmann, San Mateo, CA, 1998.
32. E. Ukkonen. Algorithms for approximate string matching. *Information Control*, 64:100–118, 1985.
33. E. Ukkonen. Approximate matching over suffix trees. In *Lecture Notes in Computer Science*, volume 684, pages 228–242. Springer Verlag, 1993.
34. B.B. Xia. Similarity search in time series data sets. Technical report, School of Computer Science, Simon Fraser University, 1997.