Multi-level Abstractions and Multi-dimensional Retrieval of Cases with Time Series Features

Stefania Montani[1], Alessio Bottrighi[1], Giorgio Leonardi[2], Luigi Portinale[1], and Paolo Terenziani[1]

[1] Dipartimento di Informatica, Università del Piemonte Orientale, Alessandria, Italy
[2] Dipartimento di Informatica e Sistemistica, Università di Pavia, Pavia, Italy

Abstract. Time series retrieval is a critical issue in all domains in which the observed phenomenon dynamics have to be dealt with. In this paper, we propose a novel, *domain independent* time series retrieval framework, based on *Temporal Abstractions* (TA). Our framework allows for *multi-level abstractions*, according to two *dimensions*, namely a taxonomy of (trend or state) symbols, and a variety of time granularities. Moreover, we allow for *flexible querying*, where queries can be expressed at any level of detail in both dimensions, also in an *interactive* fashion, and *ground cases* as well as *generalized ones* can be retrieved. We also take advantage of *multi-dimensional orthogonal index structures*, which can be refined *progressively* and *on demand*. The framework in practice is illustrated by means of a case study in hemodialysis.

1 Introduction

Several real world applications require to capture the evolution of the observed phenomenon over time, in order to describe its behaviour, and to exploit this information for future problem solving. In these applications, (many) process features are naturally collected in the form of time series, often automatically sampled and recorded by control instruments, as it happens e.g. in Intensive Care Unit patient monitoring [20], or in hemodialysis [17].

Case-based Reasoning (CBR) [1] is recently being recognized as a valuable knowledge management and decision support methodology in these domains, as testified by the relatively wide number of works in the field (see section 5). However, adopting CBR is typically non trivial in these situations, since the need for describing the process dynamics impacts both on case representation and on case retrieval, as analysed in [16]. In particular, similarity-based time series retrieval has to be addressed and optimized.

In the literature, most of the approaches to similarity-based time series retrieval are founded on the common premise of dimensionality reduction, which also simplifies knowledge representation (see the survey in [9]). Dimensionality is often reduced by means of a mathematical transform able to preserve the distance between two time series (or to underestimate it). Widely used transforms are the Discrete Fourier Transform (DFT) [2], and the Discrete Wavelet Transform (DWT) [7]. Another well known methodology is Piecewise Constant

L. McGinty and D.C. Wilson (Eds.): ICCBR 2009, LNAI 5650, pp. 225–239, 2009.

Approximation (PCA) (see e.g. [12,13]), which 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). Retrieval then works in the transformed time series space, and, with respect to the non-technical end users (e.g. the physicians), it seems to operate in a black-box fashion: the users just have to input the query, and to collect the retrieved cases, but do not (need to) see (and might not understand the meaning of) the transformed time series themselves.

In the Artificial Intelligence (AI) literature, a well known methodology is Temporal Abstractions (TA) [27,3,21,15]. TA, among the other things, have been employed for:

1. reducing time series dimensionality;
2. supporting a flexible description of phenomena at different levels of time granularity (e.g. hours, minutes, seconds);
3. providing a knowledge-based interpretation of temporal data.

Rather interestingly, TA have been scarcely explored in the CBR literature (see section 5). On the other hand, as we will extensively explain in the following, we propose to widely resort to this methodology, both for a data preprocessing step, in which time series dimensionality is reduced (see item 1 above), and as a means for supporting multiple time granularities abstractions (see item 2), at the data structure level as well as at the query level.

As regards item 1, in particular, 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 such TA methods is to move from a *point-based* to an *interval-based* representation of the data [3], where: the input points (*events* henceforth) are the elements of the discretized time series, and 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 [3]. Basic TA 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, in the time series. Through basic TA, a time series is therefore converted into a string of symbols, each one corresponding to an interval of raw data, and representing the (state or trend) value persistent over such an interval. Of course symbols can be mapped to intervals of different length (but a minimum time granularity is typically defined).

Despite the fact that TA are not as popular as the mathematical methodologies for reducing time series dimensionality, we believe they represent a valuable alternative with respect to more classical techniques in many domains (e.g. in medical or financial domains, in which TA methods are indeed well known), especially when: (i) a more *qualitative* abstraction of the time series values, as the one coded by abstraction symbols, is needed/sufficient; (ii) a user-friendly mapping between raw and transformed data has to be made available. Since the output of the TA process is a sequence of symbols, usually easier to interpret for the end user with respect to the one of a mathematical transform, which would require the implementation of an additional explanatory component, we suggest that it would be useful to calculate further levels of user-interpretable abstractions over such sequence, according to two *dimensions* (see also [29]), namely: (i) a symbol taxonomy, and (ii) a time granularity taxonomy. Actually, symbols can be organized in a taxonomy, in order to provide different levels of detail in the description of episodes (of e.g. states or trends). For instance, a taxonomy of trend symbols can be introduced (see figure 1), in which the symbol I (increase) is further specialized into I_W (weak increase) and I_S (strong increase), according to the slope. On the other hand, time granularities allow one to describe episodes at different levels of temporal detail, which is the form of TA described as item 2 above. For instance, a series of three adjacent episodes of I, I and S (stationarity), each one with a duration of 1 hour, can be merged into a single I episode, with a duration of 3 hours.

Stemming from these considerations, we are developing a *domain independent* framework for supporting time series retrieval, in which we work on cases with time series features, pre-processed by means of basic TA (henceforth *TA-based time series*), and stored in a database[1]. On such data, we support *multi-level abstractions*, i.e. abstractions at different detail levels according to the two dimensions outlined above. Moreover, we allow for *flexible querying*, where queries can be expressed at any level of detail in both dimensions, also in an *interactive* fashion, and *ground cases* as well as *generalized ones* (i.e. cases with features abstracted at a higher detail level, see section 3) can be retrieved. In our opinion, such flexibility and interactivity represent an additional advantage of TA-based time series retrieval with respect to more classical techniques, in which end users are unable to intervene in the retrieval process. Our framework takes advantage of *multi-dimensional orthogonal index structures*, which can be refined *progressively* and *on demand*, and which allow for early pruning and focusing during the retrieval process.

The paper is organized as follows. Section 2 introduces the data structures and functions which are needed to implement multi-level abstractions and to calculate distances for supporting flexible querying. Section 3 introduces our multi-dimensional index structures, and section 4 presents index definition and navigation algorithms, illustrating them by means of an example, taken from the hemodialysis domain. Section 5 introduces some comparisons with related work. Finally section 6 is devoted to conclusions and future work.

2 Data Structures and Functions for Multi-level Abstractions and Flexible Querying

As anticipated in the Introduction, we support *multi-level abstractions*, i.e. abstraction at different detail levels, according to two dimensions: (i) a taxonomy of symbols, and (ii) a taxonomy of time granularities.

[1] For the sake of clarity, in our description we will focus on cases with a single feature.

One of the main goals of our approach is generality: we aim at proposing a methodology that, in principle, can be applied to any domain in which time series are used and TA output is of interest. In particular, we want to allow maximal flexibility both in the description of the domain (i.e., in the taxonomy of symbols, and in the *distance* function measuring distances between symbols) and in the accuracy of temporal information (i.e., in the taxonomy of time granularities, and in the function for scaling up from a granularity to a coarser one - called *up* henceforth). On the other hand, we aim at assuring the "consistency" of the different descriptions. To do so, we have identified a set of general "consistency" constraints, that any meaningful choice must satisfy. Such constraints are motivated and illustrated below, within a description of the data structures for supporting multi-level abstractions (i.e. the taxonomies) and of their properties.

It is also worth noting that our approach allows to manage and integrate domain knowledge, when available, basically in the form of additional abstraction levels, both in dimension (i) and (ii) above (see figures 1 and 2 below for an example). However, also in absence of domain knowledge, our approach is applicable, since it can be reduced to a classical TA-based approach with one-level (i.e. flat) taxonomies in the worst case.

The **symbol taxonomy** is a conventional *isa* taxonomy that allows to describe the domain (states or trends) at increasingly more abstract levels of detail, starting from the bottom level, provided by the preprocessing TA step. An example taxonomy of symbols for trend TA is the one illustrated in figure 1. Of course, depending on the application domain, the tree can become wider or higher. The overall set of symbols in the taxonomy composes the *symbol domain*.

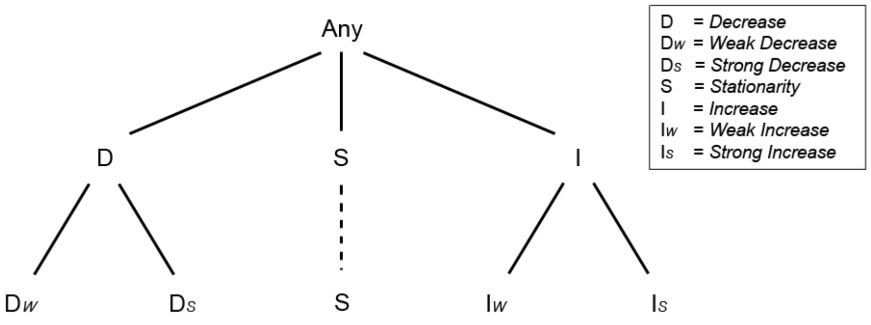

Fig. 1. An example symbol taxonomy

An important property of many symbol domains is **ordering**[2]. Such an ordering naturally emerges from the interpretation of the underlying row data, from which bottom symbols in the taxonomy have been abstracted. For instance, D_S

[2] Our framework allows to treat both ordered and unordered domains. However, we will focus on ordered ones, since ordering imposes additional constraints on the domain description which are unnecessary otherwise.

(strong decrease) may abstract curve portions with slopes from −90 to −45 degrees, thus preceding D_W (weak decrease) (referring e.g. to slopes from −44 to −10 degrees), in the symbol domain ordering. Henceforth, we will use the symbols $<^d$ and $\leq^d$ to denote (strict) precedence in the symbol domain.

Of course, the symbol taxonomy must respect the ordering (see also [4,23]), if any, as stated by the following axiom:

$$\forall x, y, x', y' \in D_s \;\; isa(x, x') \wedge isa(y, y') \wedge x' \neq y' \wedge x <^d y \rightarrow x' <^d y' \qquad (1)$$

where D_s is the symbol domain, x is a child of x', and y is a child of y' in the isa taxonomy. For instance, if D_W preceds I_W in the trend symbol ordered domain, then also D must preceed I.

A **distance** function may be used in order to measure the distance between symbols in the taxonomy. As regards the distance function choice, any one can be selected. We just enforce the staightforward general constraint that the distance of each symbol from itself is zero:

$$\forall x \in D_s \;\; d(x, x) = 0 \qquad (2)$$

where D_s is the symbol domain, and $d(x, x)$ denotes the distance between two identical symbols x.

While we do not impose any further constraint on the distance function for unordered symbol domains, we enforce the fact that distance must be "consistent" with ordering (if any). Specifically, distance monotonically increases with ordering, as requested by the following axiom:

$$\forall x, y, z \in D_s \;\; x <^d y <^d z \rightarrow d(x, y) < d(x, z) \qquad (3)$$

where D_s is the symbol domain, and $d(x, y)$ denotes the distance between symbol x and symbol y.

For instance, referring to the trend symbol domain in figure 1, where the ordering is naturally given by the increasing slope values, axiom 3 states that the distance between D and S must be smaller than the distance between D and I.

The **granularity taxonomy**, on the other hand, allows one to describe the episodes at increasingly more abstract levels of temporal aggregation, starting from the bottom level provided by the preprocessing TA step (see figure 2 for an example). Obviously, the number of levels and the dimension of granules can be differently set depending on the application domain. Observe that the time dimension requires that aggregation is "homogeneous" at every given level, in the sense that each granule at a given level must be an aggregation of exactly the same number of consecutive granules at the lower level (while this number may vary from level to level; for instance, two 30 minutes long granules compose a 1 hour long granule, while three 10 minutes long granules compose a 30 minutes long granule). Such an "homogeneity" restriction is motivated by the fact that, in such a way, the duration of each episode is (implicitly) represented in the sequence of symbols. For example, at the time granularity level of 10 minutes,

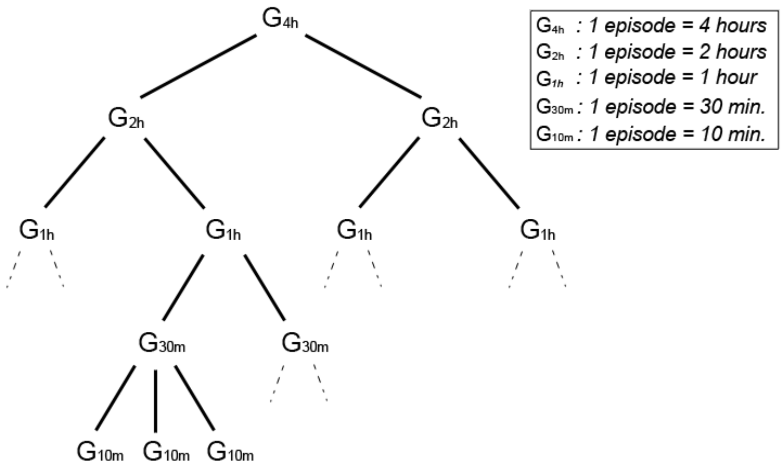
G4h
G2h
G2h
G1h
G1h
G1h
G1h
G30m
G30m
G10m
G10m
G10m
G4h : 1 episode = 4 hours
G2h : 1 episode = 2 hours
G1h : 1 episode = 1 hour
G30m : 1 episode = 30 min.
G10m : 1 episode = 10 min.

Fig. 2. An example time granularities taxonomy

the string $IIISDD$ may represent a 30 minutes episode of I followed by 10 minutes of S and 20 minutes of D.

In order to abstract along the temporal dimension, a function for **scaling up** from one level to the coarser one in the taxonomy must be provided (called *up* henceforth). Abstracting from one granularity to a coarser one is a highly domain-dependent procedure. In order to retain the maximal generality, our framework allows one to freely define the rule. Once again, however, we impose some very general constraints, to grant for the meaningfulness of the function and for its "consistency" with respect to the other knowledge sources. The following axiom grants the fact that *up* preserves "persistence": the result of coarsening two granules with the same symbol x is a larger granule still labeled as x. Here and in the following, for the sake of simplicity and brevity we apply the *up* function to two granules (but the definitions can be generalized to n-ary *up* operators):

$$\forall x \in D_s \;\; up(x, x) = x \tag{4}$$

where D_s is the symbol domain, and $up(x, x)$ denotes the symbol obtained by abstracting two adjacent intervals, both labelled with the same symbol x, at a coarser time granularity.

On the other hand, the two following axioms state the relationships between ordering and *up*, enforcing a sort of "monotonicity": in some sense, they state that ordering is preserved by the *up* function. In particular:

$$\forall x, y \in D_s \;\; x <^d y \rightarrow x \leq^d up(x, y) \leq^d y \tag{5}$$

where D_s is the symbol domain, and $up(x, y)$ denotes the symbol obtained by abstracting two adjacent intervals, labelled with the symbols x and y respectively, at a coarser time granularity. Moreover:

$$\forall x, y, z \in D_s \quad x <^d y <^d z \rightarrow up(x, y) \leq^d up(x, z) \tag{6}$$

where D_s is the symbol domain.

Given such axioms, some unclear (or, more precisely, meaningless) situations are automatically ruled out. For instance, it can never happen that, if a 1 hour long episode of D, followed by a 1 hour long episode of I, abstracts to a 2 hours long episode of D, it also happens that a 1 hour long episode of D, followed by a 1 hour long episode of S, abstracts to a 2 hours long episode of S.

It is worth stressing that the axioms above code the relationships between the symbol ordering (if any) and the *isa* relation, the distance function, and the *up* function respectively. As a consequence, the combination of such axioms also fixes the constraints between any "combination" of such primitive notions. For instance, axioms 1 and 3 state that distance "preserves" ordering also in case *isa* relationships between symbols are involved.

3 Multi-dimensional Index Structures for Retrieval Optimization

Although the use of a symbol taxonomy and/or of a temporal granularity taxonomy has been already advocated in other works (e.g. in a data warehouse context, see [29]), to the best of our knowledge we are proposing the first approach attempting to fully exploit the advantages of taxonomical knowledge in flexible case retrieval (see section 5).

Our basic idea is simple. Given the symbol taxonomy (which directly induces the abstraction function defined by the *isa* relation), the time granularity taxonomy, and the time granularity abstraction function *up*, any query can be easily abstracted at any level of symbol and/or time granularity detail (coarser than the level of the query itself). Therefore, if we provide a multi-level indexing structure addressing the different levels of abstraction, we can easily use it in order to focus our search. Starting from the most abstract level of detail, and comparing the abstracted query to the index structure nodes at progressively more accurate detail levels, our methodology can efficiently provide an early pruning of all the cases that are addressed by intermediate index layers which do not match with the query abstractions (further details on query answering will be provided in section 4).

In particular, we advocate the introduction of a forest of index structures, providing a flexible indexing of cases at different levels of the symbol and/or time granularity taxonomies. The root node of each index structure is represented by a string of symbols, defined at the highest level in the symbol taxonomy (i.e. the children of "Any", see figure 1) and in the time granularity taxonomy. Potentially, a whole taxonomy of nodes can stem from each root, describing each possible refinement along the symbol and/or time granularity dimension. An example, taking as a root the D symbol, is provided in figure 3. Here, the root node D is refined along the time dimension from the 4 hours to the 2 hours granularity, so

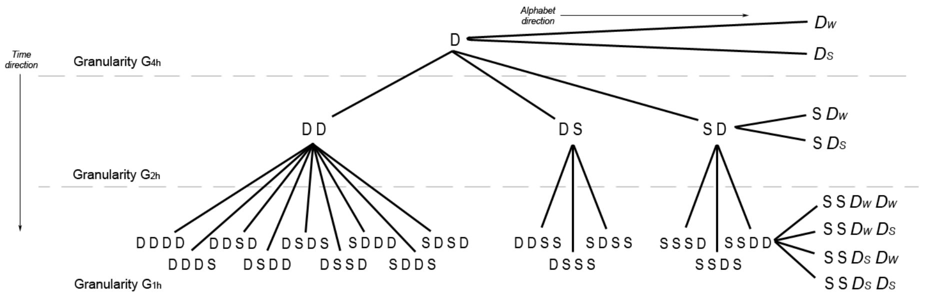

Fig. 3. An example multi-level orthogonal index structure

that the nodes DD, DS and SD stem from it, provided that $up(D, S) = D$ and $up(S, D) = D$ (see figure 3).

Moreover, we advocate that each node in each index structure belonging to the forest is itself an index, and can be defined as a *generalized case*, in the sense that it summarizes (i.e. it indexes) a set of ground cases. In the indexed cases, the feature can be abstracted as in the internal node itself (i.e. resulting in the same string), provided that we work at the same time granularity and symbol taxonomy level of the node being considered.

This means that the same ground case is typically indexed by different nodes in one index (and in the other indexes of the forest). As we will see in section 4, this supports flexible querying, since, depending on the level at which the query is issued, one of the nodes can be more suited for providing a quick answer.

Although the full generation of the forest of index structures, considering each possible level of symbol and granularity detail, is theoretically possible, for the sake of efficiency we advocate:

- the choice of a leading dimension, i.e. of a fixed order for abstractions (e.g. time granularity abstractions first, and then symbol abstractions);
- a dynamic generation/refinement of indexes, starting from a basic set of skeletal indexes which has to be defined in each specific domain/application (note that it makes sense to provide at least indexes at the coarsest symbol and time granularity levels as an initialization).

The choice of a leading dimension allows to quite naturally organize the index structure in an orthogonal way, in which, from each node of the leading dimension structure, another index stems, built according to the secondary dimension (see figure 3). In particular, the orthogonal index takes the leading index node as a root, and then progressively specializes it in the secondary dimension, keeping the leading dimension abstraction level always fixed.

It is important to stress that, although in principle the choice of a fixed order for abstractions lets our methodology loose some degree of flexibility, it does not in any way affect the expressiveness of our index structures, since, in principle, all levels of $\langle symbol, time-granularity \rangle$ detail can be coped with (just the order in which levels are organized is affected). On the other hand, such a strategy makes the process of abstracting queries and searching for the corresponding indexes much easier and faster, since an a priori fixed order of abstraction and search can be exploited. In particular, in figure 3 and in the rest of the paper, we have chosen time as the leading dimension.

4 Index Generation and Navigation

As already observed, we advocate a progressive and on-demand definition of the index structures. In particular, in the beginning it makes sense to provide a forest of trees, composed by skeletal indexes, each one rooted at a set of symbols, at the coarsest detail level, in both dimensions. Such indexes develop in the leading dimension (i.e. in time in our current approach), and are as much detailed as the domain knowledge suggests.

Further index refinement can then be automatically triggered by the types of queries which have been issued so far.

Ground queries can be answered by resorting to our abstraction mechanism and index structures. Moreover, we are able to easily treat non-ground queries as well. We just ask that all symbols in the query are at the same time granularity[3].

If queries have often involved a time granularity which is not yet represented in the index(es), the corresponding level can be created. A proper frequence threshold for counting the queries has to be set to this end. We proceed analogously by creating an orthogonal index from each node which fits the frequent queries time granularity, but does not match their symbol taxonomy level.

This policy allows to augment the indexes discriminating power only when it is needed, while keeping the memory occupancy of the index structures as limited as possible.

We will now illustrate query answering in our approach, by means of an example, taken from the hemodialysis domain. In order to highlight the most innovative features of our approach, we will show an example of a non-ground query.

Hemodialysis is the most widely used treatment for End Stage Renal Disease, a severe chronic condition which, without medical intervention, leads to death. Hemodialysis relies on a device, called hemodialyzer, which clears the patient's blood from catabolites, to re-establish acid-base equilibrium and to remove water in excess. On average, hemodialysis patients are treated for four hours three times a week. Each single treatment is called a hemodialysis session, during which the hemodialyzer collects several variables, most of which are in the form of time series. Considering a case as a hemodialysis session, we want to query the case base to search for similar cases, having preprocessed the time series by means of TA.

[3] On the other hand, queries with symbols at different levels in the symbol taxonomy dimension can be easily dealt with in our approach. In particular, it is sufficient to translate every symbol at the lowest level present in the query, thus obtaining a set of queries equivalent to the original one, but easily indexable. The logic or of the single queries results has finally to be calculated.

In particular, we will focus on a single case feature, for the sake of clarity: namely, diastolic pressure. Diastolic pressure is a very powerful indicator for evaluating water reduction from the patient's blood during a session. The reduction of water from the blood during the haemodialysis session causes a constant decrease of the blood pressure. This behaviour is correct and, even if it can sometimes cause minor problems to the patient (e.g. light head spinning), it is necessary to achieve a good water and metabolites reduction. However, in certain conditions (in particular for patients suffering from cardiovascular diseases), the reduction of water is not constant, but can be characterised by stationarity periods and sudden increasing or decreasing trend episodes. In particular, problems arise when the pressure remains stationary for the most of the time (at least half of the session), which means that no water reduction takes place. Then (sharp) decreasing episodes take place, destabilising the cardiovascular system of the patient, causing problems such as faints or collapses.

An example query summarizing this negative situation is the following: SSD_SD_W, where each symbol represents a 1 hour long episode (thus globally covering the overall 4 hours duration).

We will now show how such a query can be answered, by taking advantage of the orthogonal index structure.

Generally speaking, to answer a query, in order to enter the index structure, we first progressively generalize the query itself in the symbol taxonomy direction, while keeping time granularity fixed. Then, we generalize the query in the time dimension as well. Following the generalization steps backwards, we can enter one of the indexes in the forest from its root, and then descend along it, until we reach the node which fits the original query time granularity. If an orthogonal index stems from this node, we can descend along it, always following the query generalization steps backwards. We will stop when we reach the same detail level in the symbol taxonomy as in the original query.

If the query detail level is not represented in the index, because the index is not complete, we will stop at the most detailed possible level, which, since the abstraction order is fixed, exists and can be univocally identified. We then return all the cases indexed by the selected node.

In our example, the query generalization in the direction of the symbol taxonomy generates the sequence $SSDD$; starting from the latter, the generalization in time generates the sequences: SD (2 hours long episodes), and then D (4 hours long episode). The complete generalization procedure is shown in figure 4.

The output of the generalization process allows to identify a single index structure in the forest, namely the one whose root is D (i.e. the tree shown in figure 3) as a support for a quick query answering. Matching the steps in the generalization process to the nodes in the index structure (in the time direction), we can descend through the nodes SD, and then $SSDD$. Now, we can move "horizontally" in the symbol taxonomy direction, to reach the node SSD_SD_W, which matches exactly our query. As a result, we can retrieve all the cases indexed by such a node.

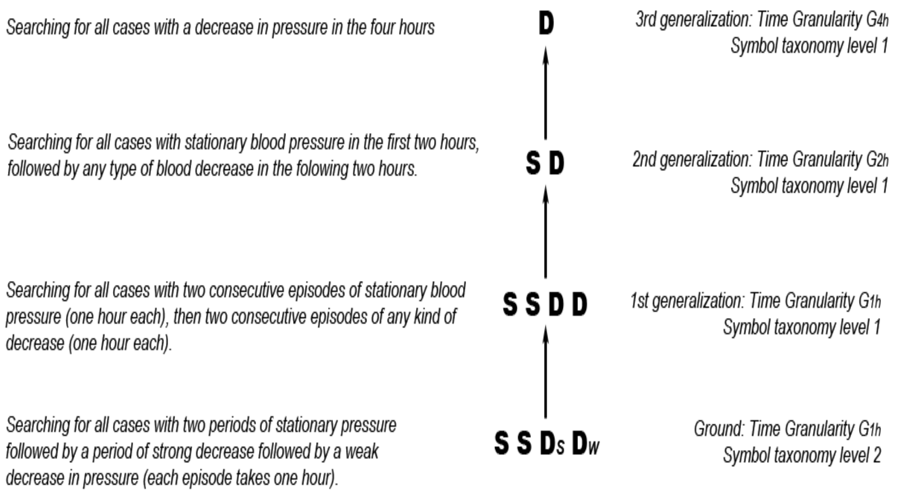
Searching for all cases with a decrease in pressure in the four hours
D
3rd generalization: Time Granularity G4h
Symbol taxonomy level 1
Searching for all cases with stationary blood pressure in the first two hours, followed by any type of blood decrease in the folowing two hours.
S D
2nd generalization: Time Granularity G2h
Symbol taxonomy level 1
Searching for all cases with two consecutive episodes of stationary blood pressure (one hour each), then two consecutive episodes of any kind of decrease (one hour each).
S S D D
1st generalization: Time Granularity G1h
Symbol taxonomy level 1
Searching for all cases with two periods of stationary pressure followed by a period of strong decrease followed by a weak decrease in pressure (each episode takes one hour).
S S DS DW
Ground: Time Granularity G1h
Symbol taxonomy level 2

Fig. 4. Generalization steps for the diastolic pressure query

Once a set of candidate cases for a given query have been selected by navigating the index structures, distance values can be calculated by introducing any distance function which satisfies the constraints illustrated by the axioms in section 2.

In our example, among the others, the case in figure 5 is retrieved[4]. As the figure shows, in such a case the pressure remains constant for approximately half of the session. Then, the two decrease episodes we were searching for take place: a strong decrease followed by a weak decrease.

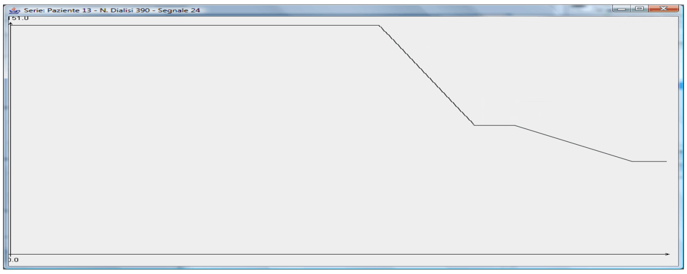

Fig. 5. Diastolic pressure in one of the retrieved cases

The query SSD_SD_W reflects a very important - but quite uncommon - situation to be investigated. Therefore, a limited number of cases are typically

[4] We are currently working on a real cases database, containing 1475 cases, belonging to 37 different patients.

retrieved by answering this query. We may want to generalize the required behaviour, in order to retrieve a larger number of cases. Interactive and progressive query relaxation and refinement are supported in our framework. For instance, we can allow any combination of decreasing episodes in the second half of the session. This can be obtained by relaxing the query in the direction of the symbols, using e.g. the sequence $SSDD$. A subsequent relaxation, compatible with the same set of situations, can be made in the direction of time, by using as a query the sequence SD (at a two hours granularity).

Query relaxation (as well as refinement) can be repeated several times, until the user is satisfied with the obtained results.

Finally, the user may want to retrieve a generalized case, i.e. to stop the search at a proper internal node in the index structure. This node subsumes a set of ground cases, but the user may just be interested in calculating the distance between the query and the node, which summarizes the retrieval set, without entering the details of all the elements composing it. For example, if the user is interested in cases with a basically stationary behaviour for the first 2 hours, and a substantially decreasing one for the following 2 hours, node SD in figure 3 can be retrieved in our framework.

5 Comparisons with Related Work

In recent years, several CBR works dealing with cases with time series features have been published, in various application domains: robot control [22], process forecast [18,24], process supervision [8], pest management [6], prediction of faulty situations [11], and medical problems [26,25,19,17]. These approaches often rely on classical mathematical dimensionality reduction techniques, such as DFT [17] and DWT [19]. Sometimes (see e.g. [25]) TA are used for data pre-processing, but basically as a noise filtering tool. Moreover, each approach has substantially been thought to support a specific application, and its generalizability is limited or not discussed at all.

A more general framework for case representation and retrieval with time dependent features has been proposed in [10]. This paper deals with the problem of time series similarity and proposes a complex retrieval strategy; we believe that our TA-based approach is more flexible, and more easily interpretable for end users. The work in [14] presents an application independent logic formalism addressing case representation when process dynamics have to be dealt with. Temporal knowledge representation for CBR is also discussed in [5]. Nevertheless, these papers do not deal with dimensionality reduction, and do not focus on retrieval solutions.

As regards TA, they have been extensively resorted to in the literature, especially in the medical field (see the survey in [28]), but typically with the aim to solve a *data interpretation task* [27] (see item 3 in the Introduction), and not as a retrieval support facility. For instance, TA have been adopted to study the co-occurrence of certain episodes in a set of clinical time series, which may justify a given diagnosis; obviously, this kind of problems are strongly based on domain knowledge, and are hardly generalizable.

Therefore, our domain independent approach for TA-based time series retrieval appears to be significantly innovative in the recent literature panorama.

It is worth noting that a database querying tool has been introduced in [29]; in this work a symbolic query (in the form of string of symbols, like the ones produced by TA) can be answered over a database of raw time series data, by producing those substrings that best match the query itself, following a set of abstraction rules, operating on a symbol taxonomy and on different time granularities. The paper thus basically introduces the same data structures we rely upon (see section 2), but exploits them only to support roll-up and drill-down operations in a data warehouse context, where the query abstraction level determines the level at which the retrieved data have to be transformed. Instead, we provide a more general and flexible retrieval support framework, in which orthogonal index structures optimize the response time, and both ground and generalized cases can be obtained. On the other hand, by now our approach operates on string matching, and not on substring matching and with the alignement problem: however, we envision such an extension as a future work.

6 Conclusions

In this paper, we have presented a domain independent framework for supporting time series retrieval, in which time series dimensionality is preliminarily reduced by means of TA. The use of TA provides an easily interpretable output, also for end users. Moreover, we support multi-level abstractions of TA-based time series, both along the time dimensions, and along the symbol taxonomy one, thus increasing the flexibility of the retrieval facility, especially in query definition. Queries, at various level of detail, can be made finer or coarser interactively. Query answering is also made faster by the use of orthogonal index structures, which can grow on demand. Indexes obviously allow for early pruning and focusing during the retrieval process.

In our opinion, flexibility and interactivity represent a relevant advantage of our approach to time series retrieval with respect to more classical techniques, in which end users are typically unable to intervene in the retrieval process, that often operates in a black-box fashion. In this work we have illustrated the framework in practice by means of a case study in hemodialysis. In the future, we plan to complete the framework implementation, and to extensively test the methodology by considering different domains, thus validating its significance, and studying ways of making it more and more efficient and usable.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations and systems approaches. AI Communications 7, 39–59 (1994)
2. Agrawal, R., Faloutsos, C., Swami, A.N.: Efficient similarity search in sequence databases. In: Lomet, D. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993)

3. Bellazzi, R., Larizza, C., Riva, A.: Temporal abstractions for interpreting diabetic patients monitoring data. Intelligent Data Analysis 2, 97–122 (1998)
4. Bergmann, R., Stahl, A.: Similarity measures for object-oriented case representations. In: Smyth, B., Cunningham, P. (eds.) EWCBR 1998. LNCS (LNAI), vol. 1488, p. 25. Springer, Heidelberg (1998)
5. Bichindaritz, I., Conlon, E.: Temporal knowledge representation and organization for case-based reasoning. In: Proc. TIME 1996, pp. 152–159. IEEE Computer Society Press, Washington (1996)
6. Branting, L.K., Hastings, J.D.: An empirical evaluation of model-based case matching and adaptation. In: Proc. Workshop on Case-Based Reasoning, AAAI 1994 (1994)
7. Chan, K.P., Fu, A.W.C.: Efficient time series matching by wavelets. In: Proc. ICDE 1999, pp. 126–133. IEEE Computer Society Press, Washington (1999)
8. Fuch, B., Mille, A., Chiron, B.: Operator decision aiding by adaptation of supervision strategies. In: Aamodt, A., Veloso, M.M. (eds.) ICCBR 1995. LNCS (LNAI), vol. 1010, pp. 23–32. Springer, Heidelberg (1995)
9. Hetland, M.L.: A survey of recent methods for efficient retrieval of similar time sequences. In: Last, M., Kandel, A., Bunke, H. (eds.) Data Mining in Time Series Databases. World Scientific, London (2003)
10. Jaczynski, M.: A framework for the management of past experiences with time-extended situations. In: Proc. ACM conference on Information and Knowledge Management (CIKM) 1997, pp. 32–38. ACM Press, New York (1997)
11. Jaere, M.D., Aamodt, A., Skalle, P.: Representing temporal knowledge for case-based prediction. In: Craw, S., Preece, A.D. (eds.) ECCBR 2002. LNCS (LNAI), vol. 2416, pp. 174–188. Springer, Heidelberg (2002)
12. Keogh, E.: Fast similarity search in the presence of longitudinal scaling in time series databases. In: Proc. Int. Conf. on Tools with Artificial Intelligence, pp. 578–584. IEEE Computer Society Press, Washington (1997)
13. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and Information Systems 3(3), 263–286 (2000)
14. Ma, J., Knight, B.: A framework for historical case-based reasoning. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS (LNAI), vol. 2689, pp. 246–260. Springer, Heidelberg (2003)
15. Montani, S., Bottrighi, A., Leonardi, G., Portinale, L.: A CBR-based, closed loop architecture for temporal abstractions configuration. Computational Intelligence (in press)
16. Montani, S., Portinale, L.: Accounting for the temporal dimension in case-based retrieval: a framework for medical applications. Computational Intelligence 22, 208–223 (2006)
17. Montani, S., Portinale, L., Leonardi, G., Bellazzi, R., Bellazzi, R.: Case-based retrieval to support the treatment of end stage renal failure patients. Artificial Intelligence in Medicine 37, 31–42 (2006)
18. Nakhaeizadeh, G.: Learning prediction from time series: a theoretical and empirical comparison of CBR with some other approaches. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) EWCBR 1993. LNCS (LNAI), vol. 837, pp. 65–76. Springer, Heidelberg (1994)
19. Nilsson, M., Funk, P., Olsson, E., von Scheele, B., Xiong, N.: Clinical decision-support for diagnosing stress-related disorders by applying psychophysiological medical knowledge to an instance-based learning system. Artificial Intelligence in Medicine 36, 159–176 (2006)

20. Palma, J., Juarez, J.M., Campos, M., Marin, R.: A fuzzy approach to temporal model-based diagnosis for intensive care units. In: Lopez de Mantaras, R., Saitta, L. (eds.) Proc. European Conference on Artificial Intelligence (ECAI) 2004, pp. 868–872. IOS Press, Amsterdam (2004)
21. Portinale, L., Montani, S., Bottrighi, A., Leonardi, G., Juarez, J.: A case-based architecture for temporal abstraction configuration and processing. In: Proc. IEEE International Conference on Tools with Artificial Intelligent (ICTAI), pp. 667–674. IEEE Computer Society Press, Los Alamitos (2006)
22. Ram, A., Santamaria, J.C.: Continuous case-based reasoning. In: Proc. AAAI Case-Based Reasoning Workshop, pp. 86–93 (1993)
23. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proc. IJCAI, pp. 448–453 (1995)
24. Rougegrez, S.: Similarity evaluation between observed behaviours for the prediction of processes. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) EWCBR 1993. LNCS (LNAI), vol. 837, pp. 155–166. Springer, Heidelberg (1994)
25. Schmidt, R., Gierl, L.: Temporal abstractions and case-based reasoning for medical course data: Two prognostic applications. In: Perner, P. (ed.) MLDM 2001. LNCS, vol. 2123, pp. 23–34. Springer, Heidelberg (2001)
26. Schmidt, R., Heindl, B., Pollwein, B., Gierl, L.: Abstraction of data and time for multiparametric time course prognoses. In: Smith, I., Faltings, B.V. (eds.) EWCBR 1996. LNCS (LNAI), vol. 1168, pp. 377–391. Springer, Heidelberg (1996)
27. Shahar, Y.: A framework for knowledge-based temporal abstractions. Artificial Intelligence 90, 79–133 (1997)
28. Terenziani, P., German, E., Shahar, Y.: The temporal aspects of clinical guidelines. In: Ten Teije, A., Miksch, S., Lucas, P. (eds.) Computer-based Medical Guidelines and Protocols: A Primer and Current Trends (2008)
29. Xia, B.B.: Similarity search in time series data sets. Technical report, School of Computer Science, Simon Fraser University (1997)