Computational Intelligence, Volume 25, Number 3, 2009

A CBR-BASED, CLOSED-LOOP ARCHITECTURE FOR TEMPORAL ABSTRACTIONS CONFIGURATION

STEFANIA MONTANI,[1] ALESSIO BOTTRIGHI,[1] GIORGIO LEONARDI,[2] AND LUIGI PORTINALE[1]
[1] *Università del Piemonte Orientale, Alessandria, Italy*
[2] *Università di Pavia, Pavia, Italy*

In the hemodialysis domain, we are implementing a case-based, closed-loop architecture aimed at configuring temporal abstractions (TA), which will be applied to time series data. The advantage of a case-based approach is the one of "quickly" obtaining a suitable TA parameter configuration, simply by looking at the most similar already configured case, where configured cases are indexed by means of contextual information. The retrieved configuration, together with the time series data, is then used as an input to a TA processing module, able to provide a set of qualitative states, trends, and significant combinations of both as an output. TA processing results can finally be evaluated, possibly leading to a (human-supervized) reorganization/revision of the case base content, to ameliorate future TA configuration sessions—thus closing the loop. The work is being integrated with RHENE, a system for case-based retrieval in hemodialysis, able to work both on raw time series data and on preprocessed (by means of TA) ones.

Key words: case-based reasoning, temporal abstractions, parameter setting.

1. INTRODUCTION

Parameter configuration is a critical issue in many artificial intelligence (AI) processes, especially when they are applied to complex domains like medical ones.

The temporal abstractions (TA) (Shahar 1997; Bellazzi, Larizza, and Riva 1998) methodology, in particular, is an AI process that requires a nontrivial configuration phase, because parameter values highly influence TA output.

TA are resorted to map large amounts of temporal information, such as the ones embedded in a time series, to a compact representation, able not only to summarize the original data themselves, but also to abstract and highlight meaningful behaviors in them. The basic principle of TA methods is to move from a *point-based* to an *interval-based* representation of the data, where: (i) the input points are the elements of the discretized time series; (ii) the output intervals (also called *episodes* henceforth) aggregate adjacent points sharing a common behavior, persistent over time. More precisely, the method described above should be referred to as *basic* TA (Bellazzi et al. 1998). Basic abstractions can be further subdivided into *state* TA and *trend* TA. *State* TA are used to extract episodes associated with *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 stability, in the time series. *Complex* TA (Bellazzi et al. 1998), on the other hand, aggregate two series of episodes into a set of higher level episodes (i.e., they abstract output intervals over precalculated input intervals). In particular, they are used to search for specific *temporal* relationships between episodes that can be generated from a basic abstraction or from other complex abstractions. The relation between time intervals can be any of the temporal relations defined by Allen (Allen 1984). 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 (e.g., a peak) in a single feature.

TA configuration usually demands for domain knowledge, which could be unavailable, or whose elicitation, exploitation and maintenance could be too time-consuming in practice.

Address correspondence to Stefania Montani, Dipartimento di Informatica, Università del Piemonte Orientale, Via T. Michel 11 I-15121 Alessandria, Italy; e-mail: stefania.montani@unipmn.it

In several applications, however, the use of knowledge about the contextual situation under examination represents an appropriate means for parameter setting; the main difficulty in these cases is the one of selecting a criterion to find the most suitable configuration from a large number of possible ones.

If we define a *case* as a set of $\langle feature, value \rangle$ pairs keeping contextual information, and store the suggested parameter configuration as the corresponding solution, case-based reasoning (CBR) (Kolodner 1993; Aamodt and Plaza 1994) can be resorted to fulfill this task. The main advantage of a CBR approach obviously stands in the fact that the knowledge acquisition process for configuring parameters is made easier by the use of already configured cases, retrieved because similar to the current input situation (Portinale et al. 2006; Montani 2008).

In this work, we propose a case-based architecture for parameter configuration of TA, to be applied to time series data, in the domain of End-Stage Renal Disease (ESRD). ESRD is a severe chronic condition that corresponds to the final stage of kidney failure. Without medical intervention, ESRD leads to death. Hemodialysis is the most widely used treatment method for ESRD; it relies on an electromechanical device, called hemodialyzer, which thanks to an extracorporal blood circuit, is able to clear the patient's blood from catabolites, to reestablish acid–base equilibrium and to remove water in excess. On average, hemodialysis patients are treated for 4 h three times a week. Each single treatment is called a hemodialysis session. Hemodialyzers typically allow users to collect several variables during a session, most of that are in the form of time series.

In our case-based approach, a suitable TA parameter configuration for an ESRD patient time series is obtained by looking at the most similar already configured case, where configured cases are indexed by means of contextual information (i.e., patient's and hemodialysis session's characteristics). The retrieved configuration, together with the time series data, is then used as an input to a TA processing module, able to provide a set of qualitative states, trends, and significant combinations of both as an output. TA processing results can finally be evaluated, possibly leading to a (human-supervized) reorganization/revision of the case base content, to improve the output of future TA configuration sessions. The configuration/processing/revision activities, are therefore, carried on within a closed-loop architecture, and realized in a semiautomatic fashion.

In the latest years we have also developed a system, called RHENE, able to retrieve hemodialysis cases, which can then be visually analyzed by physicians, or provided as an input to an automatic reasoner (for additional details, see Montani and Portinale 2006). In particular, RHENE originally worked on raw time series retrieval (just taking advantage of mathematical dimensionality reduction techniques, such as the Discrete Fourier Transform). We are currently extending it in the direction of enabling the retrieval of preprocessed time series, where preprocessing is realized by means of TA—a step that justifies the work described in this article (even though the methodology we present here is general enough to be resorted to by a tool applicable to very different domains). Actually, the closed-loop architecture introduced in this work is being integrated with RHENE.

The remaining of the article is organized as follows. In Section 2, we describe some related works. Section 3 represents the core of the contribution: in particular, in Section 3.1, we introduce case representation details; in Section 3.2, we sketch the tool's closed-loop architecture; in Section 3.3, we describe the functionality of the TA processing module, along with an example, and in Section 3.4, we analyse case base reorganization/revision. Finally Section 4 concludes the article.

2. RELATED WORK

A wide number of symbolic representations of time series have been introduced in the past decades (see Daw, Finney, and Tracy 2003 for a survey). However, most of them suffer from three main limitations: (i) the symbolic representation does not reduce dimensionality; (ii) even if distance measures can be defined on symbolic representations, these distance measures have little correlation with distance measures defined over the original time series; (iii) the conversion to symbolic representation requires to have access to all the data since the beginning.

TAs represent a possible way of dealing with some of the three issues outlined earlier. In particular (i) TA do reduce dimensionality because they convert the original point-based representation of the data to a more compact and abstract one, in which the whole time series is represented by (a few) intervals, each one describing a precise behavior (and thus easily convertible to a corresponding symbol by means of a 1:1 mapping). Moreover, (iii) they do not require to have access to the whole time series: the first TA intervals can be abstracted before the rest of the time series data is provided, thus making TA exploitable also in a context of data streaming.

It is worth citing an interesting alternative to TA, capable to deal with all issues (i)–(iii) aforementioned (Lin et al. 2003). In particular this contribution allows distance measures to be defined on the symbolic approach that lower bound the corresponding distance measures defined on the original data. Such a feature permits to run some well-known data mining algorithms on the transformed data, obtaining identical results with respect to operating on the original data, while gaining in efficiency.

Despite these advantages, the approach in (Lin et al. 2003) is not as simple as TA, which allow a very clear interpretation of the qualitative description of the data provided by the abstraction process itself. As a matter of fact, such description is closer to the language of clinicians (Stacey 2005), and easily adapts to a domain where data values that are considered as normal at one time, or in a given context, may become dangerously abnormal in a different situation (e.g., due to disease progression or to treatments obsolescence). The ease at which knowledge can be adapted and understood by experts is an aspect that impacts upon the suitability and the usefulness of intelligent data analysis systems in clinical practice. Due to these characteristics, TA have been largely exploited to support intelligent data analysis in different application areas (from diabetes mellitus Shahar and Musen 1996; Bellazzi et al. 2000; Seyfang, Miksch, and Marcos 2002, to artificial ventilation of intensive care units patients Miksch et al. 1996; Belal et al. 2005; Dojat et al. 1997; see also Stacey and McGregor 2007 for a survey).

However, while TA output can be easily understood and analysed by end users as well, without the support of a knowledge engineer or of an automatic interpretation process, parameter configuration still remains a complex task. As far as we know, not much has been done in the field of (semi)automatic parameter configuration for TA (Stacey and McGregor 2007), and usually parameter values are simply set by hand, sometimes requiring a repetition of the configuration activity until acceptable results are obtained. Remarkably, the works in (Carrault et al. 2003; Silvent, Dojat, and Garbay 2005) use machine learning techniques for the gathering of knowledge to aid in the data analysis process. However, they do not provide the ability for automatic translation and integration of that knowledge into the TA mechanisms. Therefore, our approach, which proposes to exploit CBR to support a semiautomatic and quicker parameter configuration procedure, and couples a direct interaction with experts with an automatic way of learning knowledge from data (see Section 3.4 for details), appears to be a significant contribution in the literature panorama.

3. CBR-BASED TEMPORAL ABSTRACTIONS CONFIGURATION

In this section, we provide the details of our approach. In particular, we start by quickly recalling what we mean by *case*, and how cases are represented, a point that is preliminary to the whole system description, and which was already presented in (Portinale et al. 2006) (see Section 3.1). In Section 3.2, we sketch the overall closed-loop architecture of the system, which is a revised and improved version of the one presented in (Portinale et al. 2006). Among the three modules composing the architecture, the one for TA parameter *configuration* was already described in (Portinale et al. 2006). On the other hand, the other two modules represent the main original contributions of this article. In particular, (i) in Section 3.3, we provide a description of the TA *processing* module functionality, together with a case study, that is then used as a running example through the article; (ii) in Section 3.4, we address the issue of case base *reorganization/revision*, showing how we propose to tackle two problems that could affect the output of the TA process: *missing abstractions* and *low-quality output*. The implementation of the reorganization/revision module is still ongoing, but the running example is exploited to better explain how it is meant to work.

3.1. Case Representation

As anticipated in Section 1, in TA processing the configuration phase is very important because TA output heavily depends on the chosen parameter values. In particular, for trend TA, the following (main) parameters need to be set (Bellazzi, Larizza, and Lanzola 1999): Minimum/Maximum Rate (i.e., the minimum and maximum slope allowed for the trend episode); Minimum/Maximum Duration (i.e., the minimum and the maximum duration in time for the trend episode). As regards state TA, on the other hand, we need to specify (Bellazzi et al. 1999): Lower/Upper Bound (i.e., the lower and upper bounds of data values allowed for the state episode); Minimum/Maximum Duration (defined earlier).

In our case-based approach for TA-configuration, a case is defined as follows (Portinale et al. 2006): (i) problem description: the *context description*, composed by patient and hemodialysis session characteristics that tend to be stable in the long/medium run (such as patient's age and session duration; see Section 3.3 for an example); (ii) case solution: the *configuration* of the various signals (i.e., of the time series variables collected by the hemodialyzer). In turn, the configuration of each signal consists of a list of state and trend TA symbols to be searched for in the time series to which the configuration refers, together with the corresponding parameter (i.e., Rate, Duration and/or Bound) values (see the example in Section 3.3). Optionally, a list of suitable combinations of the obtained states and trends, known as *joint TA* template, can be specified. A joint TA is a special case of a complex TA; indeed if t is an instance of a trend having validity in the time interval I_t and s is an instance of a state having validity in the time interval I_s, then $j = \langle t, s \rangle$ is a complex TA based on the Allen's relation $t\mathcal{R}s$ in the time interval $I_j = I_t \cap I_s$, where $\mathcal{R}$ is any of the following: *overlaps, during, starts, finishes, equal* and their inverse relations (Allen 1984). If the list of joint TA is empty, all $\langle trend, state \rangle$ pairs will be calculated.[1]

The initial case base has been set up with cases derived from the medical knowledge provided by a specialist.

An input case contains, together with the *context description*, a set of raw time series, instances of the signals on which TA must be extracted.

[1] Joint TA would be useful also for combining patterns abstracted from different signals; this possibility will be considered as a future work.

3.2. System Architecture

The system is conceived as a three-module architecture, composed as follows (see the dashed boxes in Figure 1):

- a module for TA parameter *configuration*;
- a module for TA *processing*;
- a module for TA-based case base *reorganization/revision*.

Given an input case, in the current implementation the TA parameter configuration module, which was already described in (Portinale et al. 2006), retrieves the less distant (i.e., most similar) case,[2] with respect to the input case context description.

The similarity distance is evaluated using the Heterogeneous Euclidean-Overlap Metric (HEOM) distance function. Given two input vectors x and y, the HEOM distance is defined as follows (Wilson and Martinez 1997):

$$HEOM = \sqrt{\sum_f d_f(x, y)^2},$$

where

- $d_f(x, y) = 1$, if x or y are missing
- $d_f(x, y) = overlap(x, y)$ if f is a symbolic feature (i.e., 0 if $x = y$, 1 otherwise)
- $\frac{|x-y|}{range_f}$ if f is a numeric and continuous feature.

The retrieved configuration information, corresponding to the signals present in the input case, is extracted and passed to the TA processing module, together with the raw data. The TA processing module, whose implementation represents one of the original contributions of this work, provides a set of qualitative states, trends, and suitable combinations of both as a result, and is extensively described in Section 3.3.

Processed examples are shown to the physician, who will decide whether to store them in the *positive examples* database or in the *negative examples* ones (see Figure 1), on the basis of the *quality* of the obtained TA series (see Section 3.4 for details). Such processed examples can then be relied upon to guide case base maintenance, by suggesting how to reorganize the configuration case library, or how to tune incorrectly defined configurations, or also how to complete the knowledge embedded in the library itself.

As described in Figure 1, the three modules thus give birth to a closed-loop architecture, where parameter configurations suggested by case-based retrieval are adopted for TA processing, while the obtained TA series are evaluated to support case base reorganization/revision. Revised cases will improve future TA configuration sessions. Details about the reorganization/revision activity, whose description is the other original contribution of this work, and whose implementation is still ongoing, are provided in Section 3.4.

[2]Note that it would be possible to generalize retrieval, by selecting a set of (more than one) very similar cases, and by combining their solutions to guide TA parameter setting. In particular, we plan to evaluate different combination techniques, such as interpolation, average, or frequency-based parameter setting.

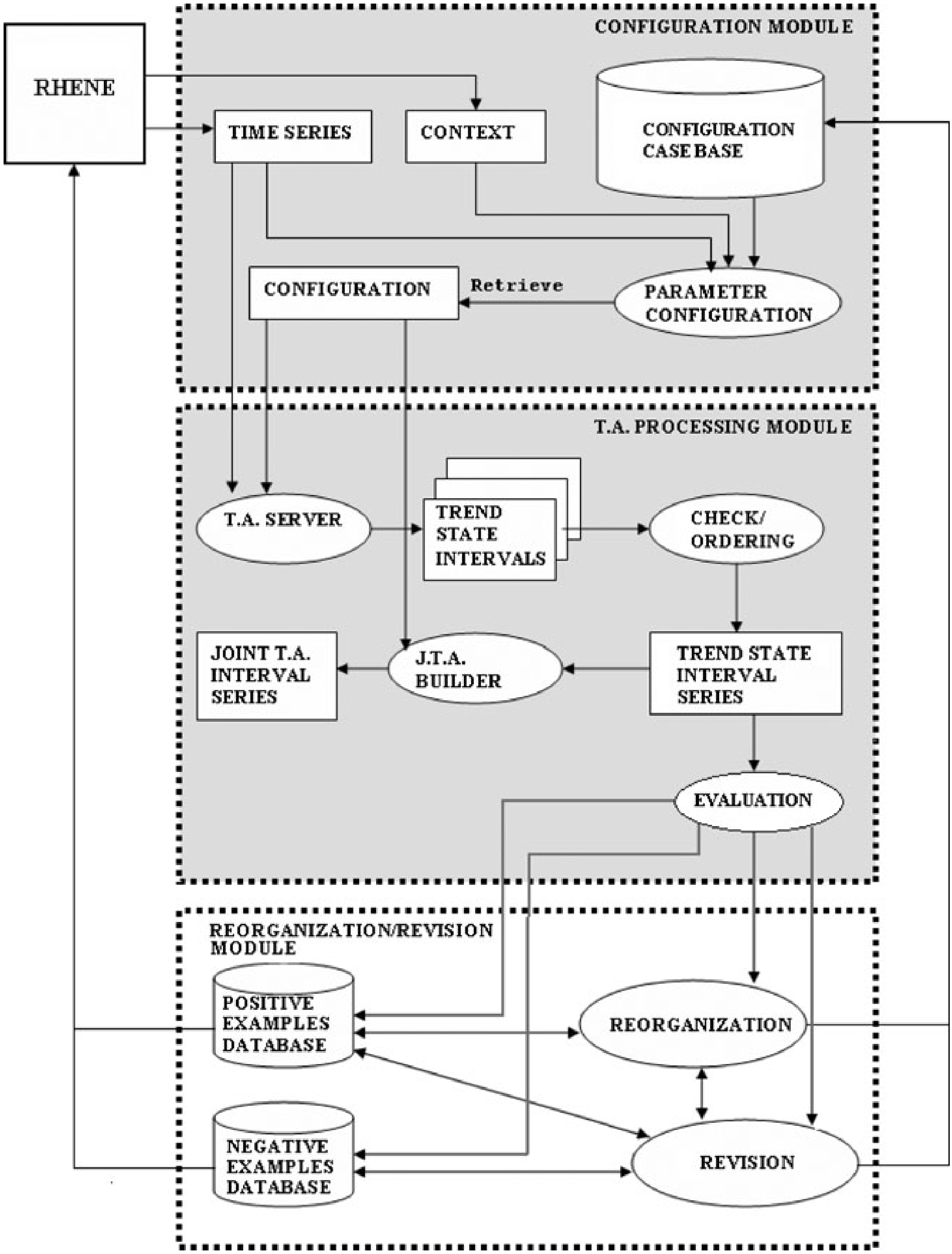

FIGURE 1. Case-based configuration of TA parameters, TA processing, and TA-based case base reorganization/revision. The dashed boxes on a gray background represent the already implemented modules, while the Reorganization/Revision module and the integration between the TA tool and the RHENE system (Montani and Portinale 2006), shown on a white background, are being realized (or will be realized as a future extension, respectively).

3.3. TA Processing

The aim of the TA processing module is to obtain a TA time series, given a raw time series with its corresponding configuration, which is provided by the configuration module (for the sake of simplicity, we are considering to deal with only one signal).

The process is articulated in three steps. In the first step, the *TA Server* (Figure 1) identifies the instances of the allowed trends and states. In our architecture, the search for predefined patterns (both trends and states) in the time series is implemented by using the TA Web service described in (Bellazzi et al. 1999). The Web service takes as an input the retrieved configuration, represented in XML format, together with the raw data to be abstracted. The output is a set of XML documents, one for each searched pattern (trend or state). Each document contains the instances of the pattern found in the data (i.e., the time intervals in which the pattern has been found).

In the second step, the *Check/Ordering* submodule in Figure 1 collects all XML documents that are generated by the *TA Server*, and creates an ordered series of trend and state instances. This submodule manages two kind of situations: **gaps** and **overlaps**.

A gap is a time interval (in the original data) where no instance of an allowed trend/state has been found. The *Check/Ordering* submodule creates an instance of a special symbol *UT* (Unknown Trend) or *US* (Unknown State) each time a gap is found for a trend or a state, respectively.

On the other hand, overlaps occur when two or more instances of different trends/states cover the same time interval. In our approach, we allow the existence of some partial overlaps between different trends or states (e.g., between *Increasing Trend* and *Stable Trend* intervals), but we exclude situations involving the following Allen's relations: *during, starts, finishes, equals* and their inverse relations. As a matter of fact, these cases (for instance, an *Increasing Trend* during a *Stable Trend*), represent a nonconsistent output of the TA extraction process, and might have been caused by a parameter misconfiguration (see Section 3.4 for details). On the other hand, the Allen's relation *overlaps* (and its inverse *overlapped-by*) are allowed, provided that the intersection interval length does not exceed a suitable threshold.[3]

The *Check/Ordering* submodule completes its work by building the sequence of the instances of the found patterns.

In the third step, the *JTA Builder* submodule in Figure 1 calculates joint TA. This submodule takes as an input the ordered series of instances obtained so far and the joint TA template of the retrieved configuration; it then builds all the pairs of trend and state instances produced by the *TA Server*, which are allowed by the joint TA template. Each pair is associated with a time interval corresponding to the intersection of the time intervals of the two basic instances composing the joint TA.

Case study to illustrate TA configuration and processing, we will now show an example in which we will abstract two time series describing the hematic volume (HV) behavior of two different patients during two hemodialysis sessions.

The behavior of the HV variable is extremely important because it is strictly related to the water reduction from the patient's blood during the hemodialysis session. The correct behavior of this signal is composed by two phases. In the first phase the session starts, and the water is extracted from the blood at high speed, until the blood pressure equals the intra-cellular pressure of the cells in the blood. In the second phase, the water passes from the cells to the blood, and is again extracted from the blood by the hemodialyzer. During this second phase the HV decrease is stable and constant. Therefore, the correct HV behavior is

[3] The threshold is defined on the basis of medical knowledge.

composed by a first exponential decrease followed by a linear decrease until the end of the session.

This behavior can be altered if the patient is affected by particular health problems. For example, hypotension can cause two kinds of problems: the low pressure can alter the first phase, slowing down the water reduction speed and generating an inefficient linear decrease; moreover, it can cause instability and difficulties in the water extraction process. These problems can also force the health care professional to interrupt the hemodialysis session before the usual 4 hours duration.

A case describing the hypotensive context and the configuration of the parameters for this kind of situation can be defined as follows (Tisler and al 2002; Cases and Coll 2002; Lee and Marks 2005):

- Context description:
  - systolic pressure: any value below 110 mmHg;
  - diastolic pressure: any value below 60 mmHg;
  - session duration: any value below 4 hours;
  - age range: any value above 64 years;
  - nurse intervention: antihypotensive drug possibly provided during hemodialysis.
- Configuration (for the HV signal):
  - Expected Trends:
    - EXD = EXponential Decrease;
    - LD = Linear Decrease;
    - FD = Fall Decrease;
    - ST = Stable;
    - INC = Increase;
  - Expected States:
    - PS = Positive State (HV increases, while in this state, which is dangerous);
    - NS = Negative State (HV decreases, which is correct);
  - Expected Joint Symbols: no symbols specified, therefore all pairs $\langle trend, state \rangle$ are allowed.

As an example, we show the definition of the main parameters of the EXD trend and NS state.

- EXD trend:
  - Symbol = EXD
  - Minimum Rate = 0.150 pts/min
  - Maximum Rate = 0.250 pts/min
  - Minimum Duration = 10 min
  - Maximum Duration = no bound
- NS state:
  - Symbol = NS
  - Lower Bound = - infinite
  - Upper Bound = 0
  - Minimum Duration = 10 min
  - Maximum Duration = no bound

We can now apply the defined configuration to analyze and compare two different situations: a session ended with a good result and a very problematic session.

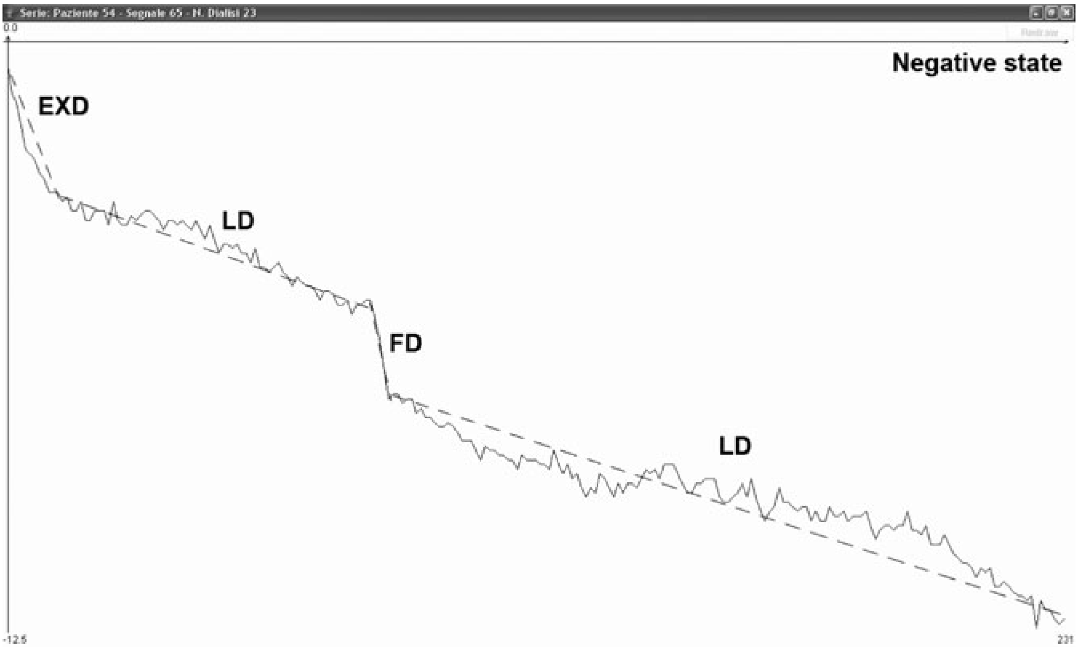

FIGURE 2. TA processing on the HV signal: a "fast reduction" example.

The first situation reports the HV in session no. 273, patient no. 54 (Figure 2) of our real examples database.[4] The state is always negative (NS), therefore the HV decreases for the entire session, while the sequence of trends (EXD, LD, FD, LD) is very near to the ideal case (EXD, LD). The episode of FD does not affect the good (i.e., fast) behavior of the water reduction.

The second case reports the HV in session no. 273, patient no. 2 (Figure 3). Here, the patient hypotension plays an important role in the water reduction. In fact, the hemodialysis session starts with an LD trend (instead of the expected EXD), followed by an INC trend that leads to a PS state. This means that the HV increases instead of decreasing. After that, the LD trend restores a NS state, but it is followed by another INC, a FD, a ST trend, then INC again and a final ST trend. Therefore, this signal presents all the problematic characteristics due to the hypotension disease, and water is reduced too slowly.

The intersection of trends and states generates the sequence of joint symbols shown in this example until the symbol J5. Considering this sequence, we can see that the symbol J1 summarizes a first interval, where the signal decreases linearly (LD), while the state is negative (NS). J2 depicts a situation where the signal increases and the state is still negative. During J3, the signal continues to increase, but the state changes from negative (NS) to positive (PS). After this increasing phase, the signal decreases again, but still in the positive state. This situation is captured by J4. The subsequent symbol U is introduced in the next interval, where no state has been recognized because the length of the two episodes are shorter than the minimum duration set for the states PS and NS. The last symbol shown in Figure 3, J5, represents a long interval where the signal decreases linearly and in the negative state.

[4]Real examples were collected at the Nephrology and Dialysis Unit of the Vigevano Hospital in Italy.

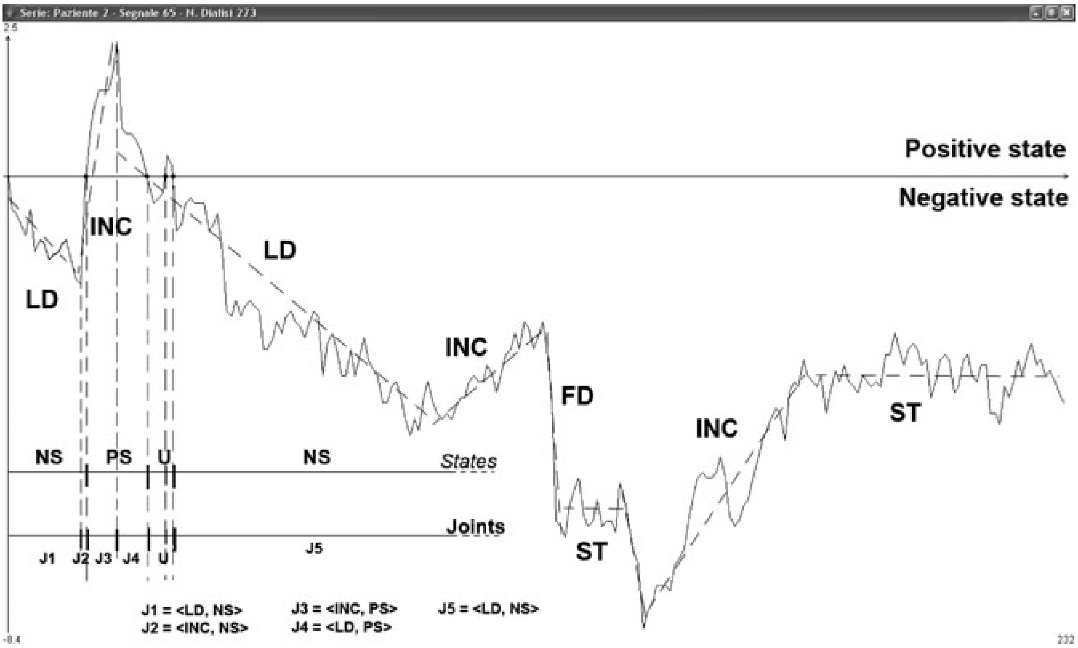

FIGURE 3. TA processing on the HV signal: a "slow reduction" example.

This sequence is very useful to perform the analysis of both trends and states together. For example, it can highlight complex episodes like the one expressed by the sequence of symbols J2 and J3: starting from the state NS, an INC of the HV leads to a PS state.

Finally observe that the analysis of the states shows an interval where no states are recognized. This is because the episodes of NS and PS that follow the correctly recognized PS are shorter then the Minimum Duration parameter; therefore this interval is marked with the symbol *US* (Unknown State).

3.4. Case Base Reorganization/Revision

The output of the TA processing module can be used to properly organize and possibly revise (in a word, to maintain) the knowledge contained in the case base. Case base maintenance is an important process in the whole CBR cycle because it may have a significant impact on the actual performance of a CBR-based system (see Leake et al. 2001 for a survey on the possible policies adopted in the literature to this end).

In particular, TA processing results may be affected by two problems, which we will call *missing abstraction* and *low quality output*, respectively. In the TA-based case base reorganization/revision module (see Figure 1), two separate submodules (*Reorganization* and *Revision* in the figure) are meant to deal with the two problems at hand. How they operate is illustrated in the two sections further.

3.4.1. Missing Abstraction. This problem is considered as a minor issue in our approach, if compared to the *low-quality output* one. Therefore, it is managed by suggesting fine-tuning of case solutions, rather than by operating a real case revision, as it happens for *low-quality output* (see Section 3.4.2).[5]

Missing abstraction can be described as follows. Suppose that a certain abstraction, or a set of (a few) abstractions, listed in the case retrieved for the input situation, has not been identified (e.g., no intervals of *Increasing Trend* could be found in time series X for the input case). This information can be exploited to enhance the case base organization, by automatically learning how to further detail context and configuration definitions, thus building a hierarchy of cases that can be resorted to realize retrieval at a finer or at a coarser granularity.

In particular, we search for past examples having the same context as the current one, in the positive examples database, and the *Reorganization* submodule (see Figure 1) verifies whether the missing abstraction problem already occurred in them. If it did, this may suggest that the configuration information (and its associated context in the retrieved case) currently groups two sets of situations (e.g., the ones in which episodes of *Increasing Trend* actually took place in time series X, and the ones in which they did not). The *Reorganization* submodule then clusters the examples, to learn what context feature values are able to distinguish between the missing abstraction set and its complement, and to properly pair the context information with the right configuration (i.e., with or without the abstraction itself). Such a distinction will allow the physician to decide whether to go in deeper detail in analysing an input situation, or not. In particular, moving toward the refined configurations may be of help to better interpret the input data, while, on other occasions, it might be helpful to remain at a coarser level and to exploit the parent configuration, to have a larger pool of comparable past examples to work with.

To illustrate how we manage the missing abstraction problem, we will resort to the hypotensive context introduced in Section 3.3. As a matter of fact, hypotension is a very frequent condition experienced by ESRD patients, therefore many examples characterized by this context can typically be collected in clinical practice. Nevertheless, as suggested by the significant differences between Figure 2 and Figure 3, a finer distinction among the hypotensive context examples can be introduced. Actually, HV in Figure 2 decreases quickly, thus water is fast reduced, while the HV behavior in Figure 3 is extremely instable and water is slowly reduced. As already observed in Section 3.3, all the abstractions indicated in the retrieved configuration are found in the "slow reduction" example, while only EXD, LD, and FD trends take place in the "fast reduction" one, and only NS is found among the expected states. This information is resorted to by the *Reorganization* submodule, which clusters the available hypotension examples, coming out with the identification of two subcontexts, one to be matched to the "fast reduction" configuration, and the other to be matched to the "slow reduction" one. The results are shown further.

- "Fast reduction" context description:
  - systolic pressure: any value between 88 and 110 mmHg;
  - diastolic pressure: any value below 60 mmHg;
  - session duration: any value between 3 hours + 10 minutes and 4 hours;
  - age range: any value between 64 years and 69 years.
- "Fast reduction" configuration (for the HV signal):
  - Expected Trends:
    - EXD = EXponential Decrease;
    - LD = Linear Decrease;
    - FD = Fall Decrease;

[5] Of course we do not exclude that the evaluation results, when available, could lead us to change this policy.

  - Expected States:
    * NS = Negative State.
- "Slow reduction" context description:
  - systolic pressure: any value below 87 mmHg;
  - diastolic pressure: any value below 60 mmHg;
  - session duration: any value below 3 hours and 10 minutes;
  - age range: any value above 70 years;
  - nurse intervention: antihypotensive drug typically provided during hemodialysis.
- "Slow reduction" configuration (for the HV signal):
  - Expected Trends:
    * EXD = EXponential Decrease;
    * LD = Linear Decrease;
    * FD = Fall Decrease;
    * ST = Stable;
    * INC = Increase;
  - Expected States:
    * PS = Positive State;
    * NS = Negative State.

The "fast reduction" and "slow reduction" $\langle context, configuration\rangle$ pairs can be separately relied upon, to perform retrieval at a finer granularity. In this way, it is possible to explicitly distinguish between situations, which despite the fact that they belong to the same context, show significant differences indeed. Retrieval at a coarser granularity, on the other hand, might be resorted to when all hypotensive examples need to be taken into account (e.g., for a comparison with an input patient not suffering from hypotension).

3.4.2. Low-Quality Output. TA output *quality* is evaluated as low if a significant number of gaps and overlaps occur.[6] Whenever an output of insufficient quality is produced by TA processing, the physician may want to accept the result as it is anyway: in this situation, the example at hand is stored in the positive examples database, and the case base content will not be updated.

Otherwise, two different situations may take place: (1) the input data are strongly affected by noise, or many missing values render the time series substantially useless for analyzing the patient's behavior: in this case the data simply have to be discarded, and the case library content does not need to be revised; (2) some problem in using the retrieved configuration has indeed emerged.

In situation (2), we store the low-quality output, obtained by having applied the retrieved case on the input data, in the negative examples database. Then, the *Revision* submodule (see Figure 1) looks for similar, past negative examples, produced by the use of the same configuration being evaluated.

If none is reported, we have identified a *competence gap* region in the configuration case base. As a matter of fact, the retrieved $\langle context, configuration\rangle$ pair always worked well, except in the present example, which is not well represented by any of the items in the library. In this situation, even though the retrieved case is the best match given the input case, their distance is typically relatively high, so that the retrieved context does not optimally describe the input one, and the corresponding configuration cannot be suitably applied. The solution is the one of acquiring a new configuration case from scratch.

[6]The *significant* number is established by the physician.

On the other hand, if some previous negative examples (i.e., examples with a low quality output) having applied the same configuration exist, we have discovered a case which was incorrectly defined since the beginning, and which needs to be revised in search of conflicting or improper parameter settings that might have been originally introduced. In this situation the *Revision* submodule also retrieves past positive examples exploiting the configuration under examination, and shows both the positive and the negative examples to the physician, to help her in the revision process.

The presence of cases having the same configuration in the two databases testifies that such configuration sometimes worked well, and in other occasions it did not. It is rather clear that a finer definition of the context features could be of help in distinguishing between these two, conflicting situations. In particular, a split of an allowed parameter range in two subsets, or the introduction of a new parameter, could accomplish this task. The physician can be supported in this revision activity by the *Reorganization* submodule clustering technique (see above), but she will be responsible for the final decision.

After the physician has completed her revision work, the new case is validated against the negative examples database: all the data over which low quality TA were extracted by using the old case are reprocessed, this time by applying the newly defined configuration. If satisfiable results are obtained, the expert will typically accept to store the new case in the configuration case base without further adjustments.

Note that both the competence gap and the misconfiguration situations require a human intervention, to define/revise the cases. As a matter of fact, in our approach case base revision is conceived as a semiautomatic procedure, to be always supervised by a domain expert. However, such a partially data-driven approach is very appealing in the hemodialysis domain, where a well-established knowledge about $\langle context, configuration\rangle$ pairs does not exist.

4. CONCLUSIONS

In this article, we have proposed a case-based architecture tackling the problem of configuring and processing TA to be applied to raw time series data, which is being implemented in the hemodialysis domain. The CBR approach does not require an explicit domain model and avoids the need of defining the right configuration for each possible contextual situation to be handled. Moreover, the CBR system can learn new knowledge by acquiring new cases or by reorganizing/revising cases which are already stored, on the basis of a detailed evaluation of the problem solving activity (TA processing in our case), in a human-supervized fashion. This gives birth to a closed-loop architecture, where (possibly revised) cases can improve the output of future TA configuration sessions.

Our work appears to be innovative in the literature panorama for at least two main reasons: (i) it is one of the very few contributions in which automatic knowledge acquisition techniques (such as CBR) are applied (Stacey and McGregor 2007), with the aim of adopting the acquired knowledge for TA processing; (ii) it moves in the direction of integrating TA processing and data analysis in a more complex architecture, in which several modules/methodologies cooperate, to better afford the challenges offered by the medical domain (Montani 2008; Stacey and McGregor 2007).

The implementation of the TA tool described in this article (see the boxes on a white background in Figure 1) is still ongoing, but we foresee to quickly complete it. As a subsequent step, we will realize the integration of the tool itself with the system RHENE, to which we will provide the capability of retrieving cases whose features are time series preprocessed by means of TA. The overall work will be followed by a testing phase on real patients' data.

To this end, we plan to provide the service to the physicians of the Nephrology and Dialysis Unit of the Vigevano Hospital in Italy.

It is worth noting that, despite the fact that the tool has been conceived within a medical application, and will be tested in the hemodialysis domain, our proposal of exploiting CBR for TA parameter setting appears to be general enough for being exported in other, nonmedical contexts. As a matter of fact, the use of knowledge about the contextual situation under examination represents an appropriate means for parameter setting in several applications. In the future, we plan to explore this possibility in more detail.

REFERENCES

AAMODT, A., and E. PLAZA. 1994. Case-based reasoning: Foundational issues, methodological variations and systems approaches. AI Communications, **7**:39–59.

ALLEN, J. F. 1984. Towards a general theory of action and time. Artificial Intelligence, **23**:123–154.

BELAL, S. Y., A. F. G. TAKTAK, A. NEVILL, and A. SPENCER. 2005. An intelligent ventilation and oxygenation management system in neonatal intensive care using fuzzy trend template fitting. Physiological Measurements, **26**:555–570.

BELLAZZI, R., C. LARIZZA, and A. RIVA. 1998. Temporal abstractions for interpreting diabetic patients monitoring data. Intelligent Data Analysis, **2**:97–122.

BELLAZZI, R., C. LARIZZA, and G. LANZOLA. 1999. An http-based server for temporal abstractions. *In* Proceedings of the IDAMAP, Washington, DC, pp. 52–62.

BELLAZZI, R., C. LARIZZA, P. MAGNI, S. MONTANI, and M. STEFANELLI. 2000. Intelligent analysis of clinical time series: An application in the diabetes mellitus domain. Artificial Intelligence in Medicine, **20**:37–57.

CARRAULT, G., M. O. CORDIER, R. QUINIOU, and F. WANG. 2003. Temporal abstractions and inductive logic programming for arrhythmia recognition from electrocardiograms. Artificial Intelligence in Medicine, **28**:231–163.

CASES, A., and E. COLL. 2002. Chronic hypotension in the dialysis patient. Journal of Nephrology, **15**:331–335.

C. DAW, C. FINNEY, and E. TRACY. 2003. A review of symbolic analysis of experimental data. Review of Scientific Instruments, **74**:915–930.

DOJAT, M., F. PACHET, Z. GUESSOUM, D. TOUCHARD, A. HARF, and L. BROCHARD. 1997. Neoganesh: A working system for the automated control of assisted ventilation in ICUs. Artificial Intelligence in Medicine, **11**:97–117.

KOLODNER, J. L. 1993. Case-Based Reasoning. Morgan Kaufmann, San Mateo, CA.

LEAKE, D. B., B. SMYTH, D. C. WILSON, and Q. YANG (eds.). 2001. Special issue on maintaining case based reasoning systems. Computational Intelligence, **17**:193–398.

LEE, D., and J. W. MARKS. 2005. Low Blood Pressure (Hypotension Definition). Available at: http://www.medicinenet.com/low_blood_pressure/article.htm. Accessed June 9, 2009.

LIN, J., E. KEOGH, S. LONARDI, and B. CHIU. 2003. A symbolic representation of time series, with implications for streaming algorithms. *In* Proceedings of ACM-DMKD, San Diego.

MIKSCH, S., W. HORN, C. POPOW, and F. PAKY. 1996. Utilizing temporal data abstractions for data validation and therapy planning for artificially ventilated newborn infants. Artificial Intelligence in Medicine, **8**:543–576.

MONTANI, S. 2008. Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. Applied Intelligence, **28**:275–285.

MONTANI, S., and L. PORTINALE. 2006. Accounting for the temporal dimension in case-based retrieval: A framework for medical applications. Computational Intelligence, **22**:208–223.

PORTINALE, L., S. MONTANI, A. BOTTRIGHI, G. LEONARDI, and J. JUAREZ. 2006. A case-based architecture for temporal abstraction configuration and processing. *In* Proceedings of the IEEE International Conference on Tools with Artificial Intelligent (ICTAI), IEEE Computer Society, Los Alamitos, pp. 667–674.

SEYFANG, A., S. MIKSCH, and M. MARCOS. 2002. Combining diagnosis and treatment using Asbru. International Journal of Medical Informatics, **68**:49–57.

SHAHAR, Y. 1997. A framework for knowledge-based temporal abstractions. Artificial Intelligence, **90**:79–133.

SHAHAR, Y., and M. A. MUSEN. 1996. Knowledge-based temporal abstraction in clinical domains. Artificial Intelligence in Medicine, **8**:267–298.

SILVENT, A. S., M. DOJAT, and C. GARBAY. 2005. Multi-level temporal abstractions for medical scenario construction. International Journal of Adaptive Control and Signal Processing, **19**:377–394.

STACEY, M. 2005. Knowledge based temporal abstractions within the neonatal intesive care domain. *In* Proceedings of the CSTE Innovation Conference, University of Western Sidney.

STACEY, M., and C. MCGREGOR. 2007. Temporal abstraction in intelligent clinical data analysis: A survey. Artificial Intelligence in Medicine, **39**:1–24.

TISLER, A., and AL. 2002. Comparison of dialysis and clinical characteristics of patients with frequent and occasional hemodialysis-associated hypotension. Kidney and Blood Pressure Research, **25**:97–102.

WILSON, D. R., and T. R. MARTINEZ. 1997. Improved heterogeneous distance functions. Journal of Artificial Intelligence Research, **6**:1–34.