A Case-Based Architecture for Temporal Abstraction Configuration and Processing

L. Portinale and S. Montani and A. Bottrighi
Dipartimento di Informatica
Universitá del Piemonte Orientale "A. Avogadro", Alessandria (Italy)
{portinal,stefania,alessio}@unipmn.it

G. Leonardi
Dipartimento di Informatica e Sistemistica
Universitá di Pavia, Pavia (Italy)
gleonardi74@virgilio.it

J.M. Juarez
Universidad de Murcia, Murcia (Spain)
jmjuarez@um.es

Abstract

In this work we propose a case-based architecture tackling the problem of configuring and processing temporal abstractions (trends and qualitative states) produced from raw time series data. The parameter configuration is a critical problem in many temporal abstraction processes; in several application domains (especially in medical ones), contextual knowledge plays a fundamental role in the time series interpretation. Since defining the right configuration for each possible contextual situation may be impractical, we propose to adopt a case-based approach, where the suitable configuration can be obtained by looking at the most similar already configured case, with respect to the current situation. Configured cases are indexed by means of contextual information. The obtained configuration can then be used as input to a temporal abstraction module, providing a set of qualitative states, trends and suitable combination of both as a result. Cases can then be exploited in the processing of such results as well, by providing an evaluation of the whole abstraction processing, possibly leading to the revision of the case base. The approach is illustrated by means of an example taken from a medical application, concerning the monitoring and evaluation of patients undergoing hemodialysis treatment.

1 Introduction and Architecture's Overview

Temporal Abstractions (TA) [11, 5] is an AI methodology able to solve a data interpretation task, the goal of which is to derive high level concepts from time stamped data. Through TA, large amounts of temporal information, such as the ones 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 behaviors in the data themselves; moreover, by means of TA, a clear mapping between raw and transformed data is made available and the mapping itself can be easily interpreted by end users as well.

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 (*events* henceforth) are the elements of the discretized time series; (ii) the output intervals (*episodes* henceforth) aggregate adjacent events sharing a common behavior, persistent over time. More precisely, the method described above should be referred to as *basic* TA [5]. 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 stationarity, from the time series. The output results of a basic TA depend on the value assigned to specific parameters, such as the admissible range of values for state TA and the slope for trend TA.

Complex TA [5] can be defined as well: instead of aggregating events into episodes, complex TA aggregate two series of episodes into a set of higher level episodes (i.e., they abstract output intervals over precalculated input intervals). In particular, complex abstractions 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 [2]. 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.

Parameter configuration is a critical issue in many temporal abstraction processes, especially when they are applied to complex domains like medical ones. The main difficulty is to select a criterion to find the most suitable configuration from a large number of possible ones. Our approach considers the fact that in several application domains, the use of knowledge about the contextual situation under examination, together with the nature of the examined time series, conform an appropriate criterion to select a proper configuration.

In this work we propose a case-based architecture [7] for the parameter configuration and the processing of temporal abstractions of time series. The main advantage of a case-based approach stands in the fact that the knowledge acquisition process for the parameter configuration task is mitigated by the use of already configured cases, which can be re-used in a similar situation; moreover, the processing of a time series using a given configuration may provide a significant feed-back for the possible revision of the used configuration (for example when the resulting temporal abstractions do not account for a significant part of the input time series), leading to the learning (i.e. addition in the case library) of new cases or to the adaptation of old ones.

Thus, we propose a two-module architecture, as follows (see Figure 1):

- a case-based module for the *parameter configuration* of the temporal abstractions.

- a module for the *processing* (creation and evaluation) of the temporal abstractions.

The first module provides a solution to the parameter configuration problem, by exploiting the context description through Case-Based Reasoning (CBR) techniques [7, 8, 1, 12]. This module is explained in detail in Section *Case-Based Configuration*.

The configuration obtained is received as input by the processing module, together with the raw time series to be abstracted. Such a module processes this information at two levels; at the first level, temporal abstraction is performed based on a classical approach of extraction of trends and states from time series (i.e. basic TA) [11, 5]. The output are interval sequences of trends and states. However, in some situations, the joint analysis of the temporal evolution of both trends and states provides an important additional information about their transitions. In these cases, we propose to use a second level of abstraction, joining trends and states, in order to understand the combined behavior (in terms of both trends and states) of the original time series. This corresponds to extract a particular kind of complex TA, based on some of the Allen's time interval relations (i.e. any

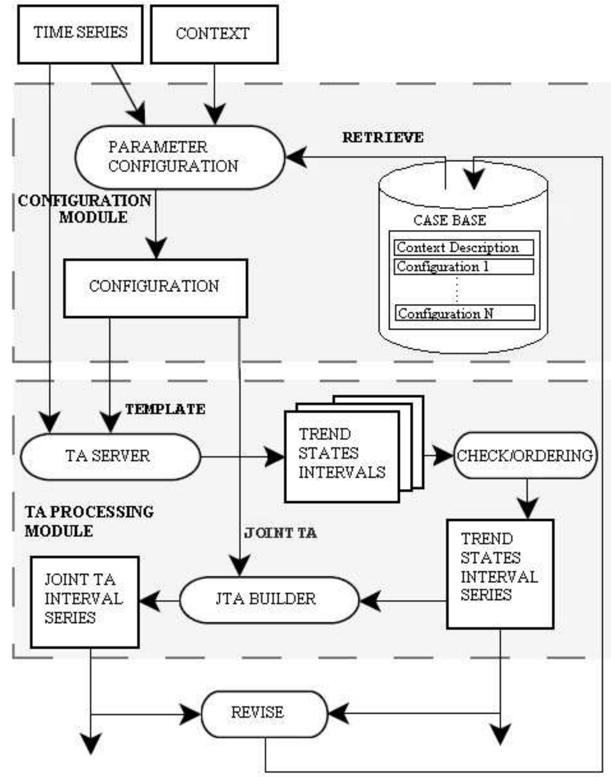

Figure 1. General Overview of the Architecture.

relation different than *before, meets* and their inverse relations) and on the intersection operator among episodes. The details of this module are described in Section *TA Processing Module*.

2 Case-Based Configuration

The aim of this module is to obtain a parameter configuration for the TA process, given the original time series and its context description. This is performed by means of a case retrieval system. In order to properly define a TA parameter configuration problem, we need to specify what kind of temporal abstractions we are going to deal with. In the following, we will consider the TA specifications supported by the TA web service described in [4], which corresponds to the *TA Server* module shown in Figure 1. As previously mentioned, we are considering two types of basic TA: **trends** and **states**. For trends it is necessary to specify the following parameters:

- *Trend*: the kind of trend (e.g. *Increasing*, *Decreasing*, *Stationary*);

- *Local Window*: the size of the sliding window used by the TA server to interpolate the signal (time series) and extract the abstractions;
- *Maximum Time Gap*: the maximum distance between two consecutive points that do not fit the trend definition, but that could be integrated in the analyzed interval by means of interpolation;
- *Minimum/Maximum Rate*: the minimum and maximum slope allowed for the trend;
- *Minimum/Maximum Duration*: the minimum and the maximum duration in time for the trend.

States are searched for using these parameters:

- *Lower/Upper Bound*: the lower and upper bounds of data values allowed for the state;
- *Local Window, Maximum Time Gap, Minimum/Maximum Duration*: defined as above.

Let us now introduce some basic definitions.

Definition 1 *An expected trend is defined as:*

$$expected\_trend = \{TrendSymbol, Trend, MaximumTimeGap, MinimumRate, MaximumRate, MaximumDuration, MinimumDuration, LocalWindow\}$$

where TrendSymbol *is a symbol that identifies the expected trend and the other parameters are defined as above.*

Definition 2 *An expected state is defined as follows:*

$$expected\_state = \{StateSymbol, MaximumTimeGap, LowerBound, UpperBound, MinimumDuration, MaximumDuration, LocalWindow\}$$

where StateSymbol *is a symbol that identifies the expected state and the other parameters are defined as above.*

The TA server uses a set of expected trends and a set of expected states to search for such abstractions in the original time series. It produces a set of instances of (expected) trends and states as follows:

$$instance\_of\_expected\_trend = \{TrendSymbol, start\_time\_point, end\_time\_point\}$$

$$instance\_of\_expected\_state = \{StateSymbol, start\_time\_point, end\_time\_point\}$$

Definition 3 *A template τ is a pair:*

$$\tau = \langle set\_of\_expected\_trend, set\_of\_expected\_state \rangle$$

Given a template τ, we will indicate as T_τ and as S_τ the set of trend symbols and state symbols respectively, which are defined in τ.

Definition 4 *Let t be a trend symbol and s be a state symbol; a pair $\langle t, s \rangle$ is denoted as a* joint TA.

Definition 5 *Given a template τ defined as above, a joint template γ_τ is defined as*[1]*:*

$$\gamma_\tau \subseteq (T_\tau \times S_\tau)$$

A template provides the *TA Server* a list of possible basic parameters to search for, while a joint template specifies which are the possible combinations of trends and states (joint TA) which have to be considered. A joint TA is actually a special case of a complex TA as defined in [5]; 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 R s in the time interval $I_j = I_t \cap I_s$, where R is any of the following: *overlaps, during, starts, finishes, equal* and their inverse relations [2]. Now, let us introduce what we mean by configuration of a TA problem. First of all, let us suppose we have to deal with a specific set of signals Σ; every signal $\sigma \in \Sigma$ represents a time-varying feature of the application under examination, each instantiation of which is a time series of raw data.

Definition 6 *A parameter configuration (or simply a configuration) for a signal σ is defined as:*

$$Configuration_\sigma = < \tau_\sigma, \gamma_{\tau_\sigma} >$$

where the first component (τ_σ) is a template associated to σ which describes the expected characteristics concerning trends and states and the second component is a joint template (γ_{τ_σ}) which contains the elements necessary for a joint analysis of the trends and states of τ.

Given the above definitions, we can now define what is in a case as well. In CBR, a case is usually assumed to be the correspondence between the situation to solve (problem description) and the set of actions to resolve it (solution) [7, 8]. In the proposed architecture, the problem description corresponds to the context description of the situation under examination, while the solution is the configuration of the TA parameters needed in order to process the signals in the described contextual situation. Therefore we state a case as follows:

[1] Notice that, since a trend/state symbol univocally defines an expected trend/state (with well specified parameters), a joint template may also be viewed as a subset of the cartesian product of the set of expected trends and the set of expected states of τ.

Definition 7

$$Case = \langle ContextDescription, \langle Configuration_{\sigma_1}, \ldots Configuration_{\sigma_n} \rangle \rangle$$

where $\sigma_1 \ldots \sigma_n$ *are the signals to be considered and* $ContextDescription$ *may be any data structure (like for instance a set of* $\langle feature, value \rangle$ *pairs) allowing for the specification of the contextual information about the current situation to be analyzed (see Section* Example: case definition and retrieval *for a concrete example).*

Def. 7 refers to cases stored in the case base, containing the set of plausible configurations for the abstractions of the signals related to the prototypical situation corresponding to the context description in the case. Of course, an input case will not contain any configuration (since configurations correspond to the case solution), but will contain, together with the context description, a set of raw time series, instances of the signals whose TA configurations must be looked for. Such time series are labeled with the signal they correspond to and are used to select the suitable configuration in the retrieved cases, before being passed to the TA processing module for the actual abstraction process.

Definition 8 *An input case is defined as:*

$$InputCase = \langle ContextDescription, \langle TimeSeries_{\sigma_{i_1}}, \ldots TimeSeries_{\sigma_{i_m}} \rangle \rangle$$

Given an input case, the parameter configuration module performs a *retrieval* on the case base as follows: a suitable notion of distance is defined among the features corresponding to the context descriptions of the cases [13]; the least distant cases (with respect to the input case) are identified and retrieved as the most similar ones to the current situation. Without lack of generality, let us suppose that the retrieval will consider only the most similar case; the configurations corresponding to signals present in the input case (i.e. $\sigma_{i_1}, \ldots \sigma_{i_m}$ in def. 8) are extracted and passed to the TA processing module, together with the corresponding raw data (i.e. $TimeSeries_{\sigma_{i_1}}, \ldots TimeSeries_{\sigma_{i_m}}$ in def. 8) (see Section *TA Processing Module*).

2.1 Example: case defi nition and retrieval

As a concrete example of a configured case, let us consider a medical domain, concerning the hemodialysis treatment. Hemodialysis is an ideal domain for the application of this architecture, since the interpretation of the biomedical signals is strictly conditioned by the context knowledge. The work in [10] reports about the RHENE system, a CBR system able to support hemodialysis therapy evaluation . The RHENE system processes and analyzes 11 signals as time series coming from the hemodialyzer (e.g. hemoglobin, systolic/diastolic pressure, hematic volume, etc ...). This system considers every signal as a feature, while the context description is composed by:

- patient description (e.g. demographic information, antecedents, treatments);
- long-term factors (e.g. tolerances on the dialysis session parameter, patient's conditions);
- session factors (e.g. dialysis session durations, blood flow, dialyzer conditions).

An important clinical problem, that may occur in this domain, is related to patients suffering from hypotension. During an hemodialysis session, a constant slight decrease of the blood pressure occurs, because of the removal of water and methabolites. This blood pressure behavior is normal, but in some cases, especially for hypotensive patients, the pressure lowers so much that the patient becomes at risk of collapse. The risk also raises with the patient's age and leads to a double problem: the collapse itself and the consequent premature interruption of the dialysis session. To avoid this occurrence, the duration of the dialysis session is usually shorter than usual (four hours for normal patients) and, in case of excessive fall of pressure, a specific drug intervention (e.g. mannithol) is performed, in order to restore a normal blood pressure state and conclude the session. A prototypical case supporting parameter configuration for hypotension context could then be defined as follows:

- Relevant Context:
  - presence of hypotension disease: systolic pressure below 110 mmHg or diastolic pressure below 60 mmHg [6];
  - session duration: 3 hours and 30 minutes;
  - age range in which the collapse is most probable: a lower bound of 64.4 [3];
  - nurse intervention with mannithol.

  For the sake of brevity, we just take into account the diastolic pressure signal in this example (one of the most interesting signal to study in this contextual situation).

- Configuration (for the diastolic pressure signal):
  - Expected Trends:
    - ST = Stable;
    - SD = Strong Decrease (to highlight the moments of blood pressure fall);

    * SI = Strong Increase (to highlight the blood increase after the nurse intervention).
  - Expected States:
    * N = Normal pressure level;
    * H = Hypotension.

- Expected Joint Symbols: No symbol specified, all pairs $\langle trend, state \rangle$ are allowed.

As an example of parameter definition[2], we consider the SD trend and the H state.

- For the SD trend, we define:
  - SymbolTrend = SD
  - Trend = Decreasing
  - MinimumRate = 85 degrees
  - MaximumRate = 88 degrees
  - MinimumDuration = 10 min.
  - MaximumDuration = no bound
- The H state is defined as:
  - SymbolState = H
  - LowerBound = 23 mmHg
  - UpperBound = 59 mmHg
  - MinimumDuration = 6 min.
  - MaximumDuration = no bound

Let us now suppose that the following dialysis session has to be analyzed as input case: the patient suffers from hypotension (initial systolic pressure starting from 105 mmHg), his age (71 years old) falls in the band of high risk of collapse, the session duration has been set to the canonical value of 4 hours. Case retrieval provides the above described case as the most similar to the input one, so the configuration retrieved for the diastolic pressure signal contains the ST, SD and SI trend definitions and the N and H state definitions. In the following, we will show how this information is exploited by the TA processing module and used to evaluate the produced abstractions, in order to possibly revise the case base.

3 TA Processing Module

The aim of this module is to obtain a temporal abstraction, given a set of raw time series with their corresponding configurations, that are provided by the configuration module. For the sake of simplicity, let us consider to deal with only one signal. In the first step, the *TA Server* (Figure 1) identifies the instances of expected trends and states. In our architecture, the search of predefined patterns (both trend and state) in the time series is implemented by using the TA web-service described in [4]. The web-service takes as input the retrieved configuration, represented in XML format, together with the raw data to be abstracted. The output is a set of XML document, 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, which are generated by *TA Server* and creates an ordered series of trend and state instances. This submodule manages two kind of situation: **gaps** and **overlaps**. A gap is a time interval (in the original data) where no instance of an expected 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. Overlaps occur when two or more instances of expected trend/state cover the same time interval. In our approach, we allow the existence of partial overlaps; in particular, situations involving the following Allen's relations are not allowed: *during*, *starts*, *finishes*, *equals* and their inverse relations. They represent situations of "temporal overlap" among intervals that are considered not consistent and are caused by errors in the specification of configuration parameters (for instance a trend of increase during a stable trend). The Allen's relation *overlaps* (and its inverse *overlaped-by*) is allowed, only if the intersection interval does not exceed a given threshold. If a configuration problem is detected, the user is asked to revise the parameters of the involved trends or states. If the analysis is considered valid, the *Check/Ordering* submodule builds the sequence of the instances of the found patterns.

The second level of abstraction is obtained by the *JTA Builder* submodule in Figure 1. This submodule takes as input the ordered series of instances obtained at the first level of abstraction and the joint template of the retrieved configurations; it then builds the corresponding series of joint TA instances. In details, the *JTA Builder* produces all the pairs of trend and state instances produced by the *TA Server*, which are allowed by the joint 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.

3.1 Example: temporal abstraction processing

Let us come back to the example of Section *Example: case definition and retrieval*. Figure 2 shows the time series corresponding to the signal diastolic pressure in the input case.

[2] We just mention here the most important ones.

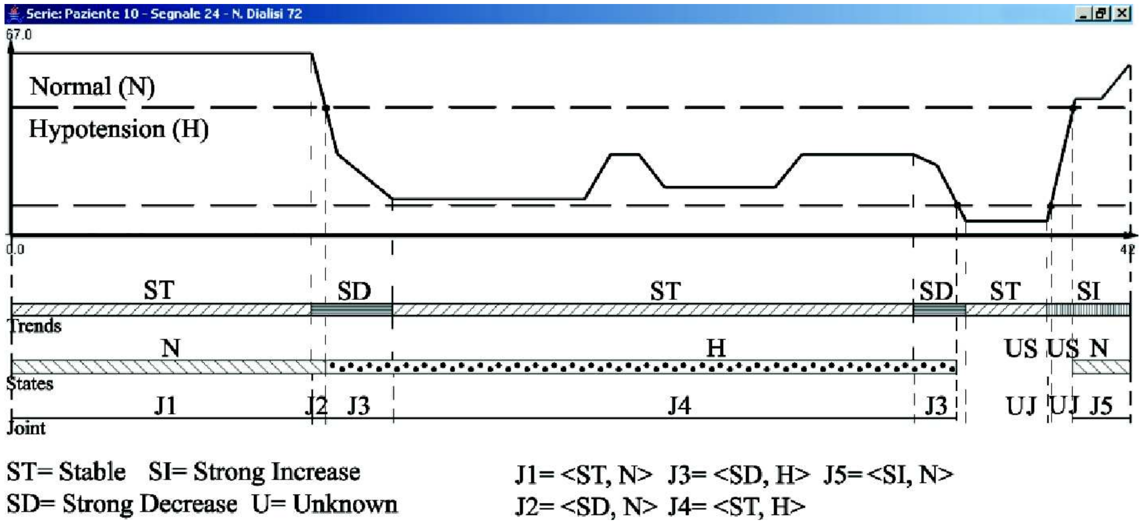

Figure 2. Analysis of a real example of diastolic pressure.

Using the retrieved configuration, the *TA Server* generates the sequences of trends and states which are reported in the figure. Looking at them, we can see that the signal starts with a $Stable$ trend and the blood pressure in $Normal$ state. Suddenly, the pressure falls down, with a $StrongDecrease$ trend, (direction coefficient of 87,8 degrees, duration of 11 minutes and 25 seconds), changing state from $Normal$ to $Hypotension$. The patient remains in this state (with Stable trend) for a long time, until the pressure falls again with a $StrongDecrease$ (direction coefficient of 85,4 degrees, duration of 11 minutes and 25 seconds). This fall leads the pressure below the lower bound value set for the $Hypotension$ state so, while the signal stays below this threshold, the process cannot recognize any state and marks this interval as US. The case also reports that, because of this situation, a drug intervention has been operated, in order to restore the correct pressure level. After this intervention, the blood pressure raises very fast with a $StrongIncrease$ trend (direction coefficient of 85,4 degrees, duration of 17 minutes and 8 seconds), returning very quickly to the $Normal$ state. Note that the time interval of the $Hypotension$ state crossed during this increase is shorter than the $MinimumDuration$ parameter set for this state (2 minutes and 51 seconds for this episode versus the 6 minutes set in the definition), so another US symbol has to be added to the state sequence.

The *JTA Builder* extracts the joint symbols sequence, starting with the symbol J1 ($Stable$ trend in a $Normal$ state). It is interesting to consider subsequences of joint symbols to discover changes in the signal that the sequence of just trends or just states cannot highlight. For example, the subsequence J2-J3 evidences the transition from $Normal$ to $Hypotension$ state during a $StrongDecrease$. This joint information is a clear signal of a situation of alarm, that is confirmed by the protraction of such a situation as indicated by J4 (a $Stable$ trend, while the patient is still in $Hypotension$). followed by J3 again, (a $StrongDecrease$ in $Hypotension$ state). After this, the presence of the US symbol imposes (by definition) to set the UJ symbol in the same interval of the joint series. J5 closes the series, showing a $StrongIncrease$ associated to a $Normal$ state and is a clear interpretation of the successful drug intervention performed to avoid collapse.

4 Evaluation and Case Base Maintenance

The output of the *TA Processing* module can be used to revise the knowledge contained in the case base. This is an important process in the whole CBR cycle, since it may affect the actual performance of any CBR-based system (see [9] for a survey on the possible case base maintenance policies). An important feed-back about the suitability of the retrieved case can be provided by an evaluation of the resulting series of TA; in particular, the occurrence of $unknown$ symbols and of TA overlaps over *long* time intervals (with *long* being a term to be defined wrt a particular application) suggests that a revision of the used configuration may be appropriate. By justifying the presence of such problems in terms of the definition of configuration parameters, the system can suggest either the adaptation of a retrieved configuration or the learning (addition in the case base) of a new case. Let us illustrate this point again through an example.

4.1 Example: evaluation and revision

Consider again our running example: as regards to the considered application, the obtained TA are affected by the presence of $unknown$ symbols for long intervals. The system can then alerts the user about this problem, trying to suggest a remedy, by considering the causes that have led to

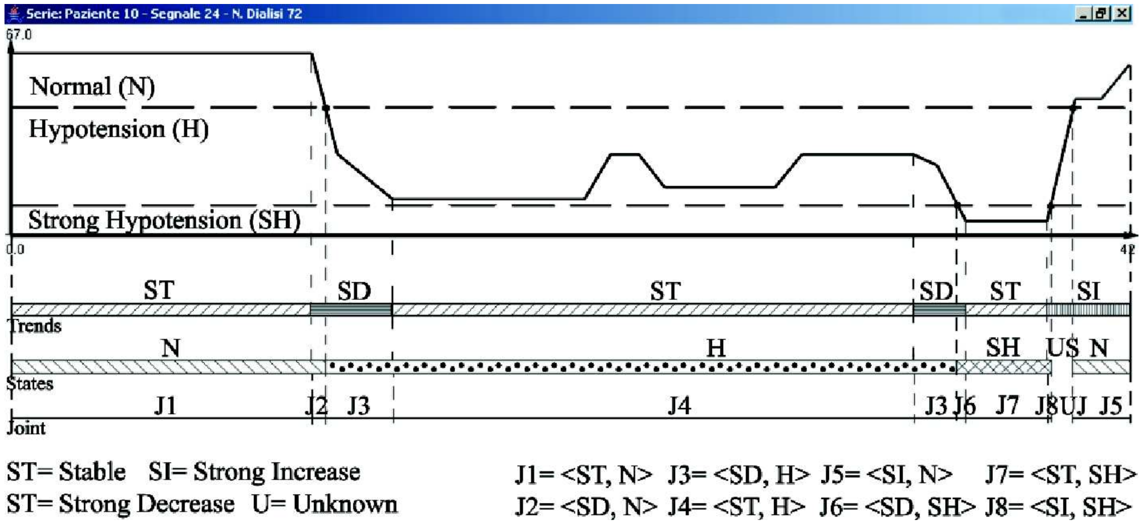

Figure 3. A second analysis of a real example of diastolic pressure.

the presence of the $unknown$ symbols. Two kinds of problem can be identified: the first is due to the lack of a state definition in the configuration template (a very low pressure level). The second one is due to a too restrictive definition of the $MinumumDuration$ parameter for $Hypotension$ state. Two different suggestions are then proposed as a solution to the user: to add a new state and to change the $MinimumDuration$ of the $Hypotension$ state. Let us suppose that the user accepts just the first suggestion: a new state is then introduced, called $StrongHypotension$ (SH), with $UpperBound = 22$ mmHg and $LowerBound = 0$ mmHg. This also leads the user to add new pairs to the joint template of the configuration (those involving the new SH state).

This system-guided evaluation of the current case can be the basis for revising the case base. Two alternatives are possible: to substitute the old configuration of the retrieved case with the new one (i.e. the addition of the SH state to the retrieved configuration), or to add a new case in the case base, corresponding to the current input case with the modified configuration. In general, if the retrieved case has a significant distance from the input one, this can be interpreted as the presence of a competence gap in the case base and the second alternative should be adopted; otherwise the first alternative appears to be more appropriate.

In this example, comparing the retrieved case with the input one, a quite significant difference can be determined in the session duration (4 hours vs 3 hours and 30 minutes); this should lead to the learning of the input case with the newly produced configuration.

Figure 3 shows the results obtained by applying the modified configuration to the input time series.

We can see no changes in the trends series, while in the states series, the new state SH is obviously recognized, following the H state. Other changes can be found in the joint TA series, with the identification of the new J6, J7 and J8 symbols, instances of the new expected joint TA symbols generated after the definition of the SH state.

This new analysis allow to highlight a more detailed information: for example, the joint TA sequence J3-J6 shows a $StrongDecrease$ which leads from $Hypotension$ to $StrongHypotension$ state. J7 tells us that the patient remains in $StrongHypotension$ state with $Stable$ trend, while the triple J8-UJ-J5 represents the $StrongIncrease$ trend with transition from $StrongHypotension$ to $Normal$ state after the drug intervention. Notice that the interpretation of the UJ symbols should in this case corresponds to the H state; however, in order to avoid to misinterpret noisy data as actual H states, parameter revision for such a state has not been performed and the transition among SH and N can be naturally accepted, because it occurs during an SI (i.e. a strong/rapid increase) trend.

5 Conclusions

In this paper, we have proposed a case-based architecture tackling the problem of configuring and processing Temporal Abstractions obtained from raw time series data. 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, a CBR system can learn new knowledge by acquiring new cases or by revising cases which are already stored, on the basis of a detailed evaluation of the problem solving activity (TA processing in our case). Of course, the possibility of identifying and extracting contextual knowldge from the application is crucial and this may not always be possible in some application fields. We have illustrated the potentiality of such an architecture, by considering, throughout the paper, a concrete medical example, where contexts are important and explicit factors: the application domain of hemodialysis. We have shown how the TA analysis of a particular signal can be usefully supported by a case-based approach. Concerning this application domain, a clinical evaluation of the architecture is currently performed under a specific research project, concerning the intelligent data analysis of the monitored data of a hemodialysis center.

6 Acknowledgement

This work has been partially supported by the italian MIUR project COFIN2004 and by the National Spanish PhD fellowship AP2003-4476. We also thank Riccardo Bellazzi and Cristiana Larizza for useful discussions about the topic of this paper and for having made available the software tool for the temporal abstraction module.

References

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

[2] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[3] A.Tisler, K. Akcsi, I. Hrshegyi, G. Varga, S. Ferenczi, M. Grosz, I. Kulcsr, L. Lcsey, J. Smik, I. Solt, J. Szegedi, E. Tth, G. Wgner, and I. Kiss. Comparison of dialysis and clinical characteristics of patients with frequent and occasional hemodialysis-associated hypotension. *Kidney Blood Press Res*, 25:97–102, 2002.

[4] R. Bellazzi, C. Larizza, and G. Lanzola. An http-based server for temporal abstractions. In *Proc. IDAMAP'99*, pages 52–62, 1999.

[5] R. Bellazzi, C. Larizza, and A. Riva. Temporal abstractions for interpreting diabetic patients monitoring data. *Intelligent Data Analysis*, 2:97–122, 1998.

[6] A. Cases and E. Coll. Chronic hypotension in the dialysis patient. *Journal of Nephrology*, 15:331–335, 2002.

[7] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.

[8] D. Leake. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press, 1996.

[9] D. Leake, B. Smyth, D. Wilson, and Q. Yang(eds). Special issue on maintaining case based reasoning systems. *Computational Intelligence*, 17(2):193–398, 2001.

[10] S. Montani, L. Portinale, R. Bellazzi, and G. Leonardi. Rhene: a case retrieval system for hemodialysis cases with dynamically monitored parameters. In *Proc. ECCBR 2004, in: Lecture Notes in Artificial Intelligence 3155*, pages 659–672. Springer, 2004.

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

[12] I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publ., 1997.

[13] D. Wilson and T. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.