TrPM: A Linguistic Petri Nets module to describe the trends of a time series

Juan Moreno-Garcia
Universidad de Castilla-La Mancha
Escuela de Ingeniería Industrial y Aeronáutica
Toledo, Spain
juan.moreno@uclm.es

Ester del Castillo
Universidad de Castilla-La Mancha
Escuela Superior de Informática
Ciudad Real, Spain
Ester.Castillo@uclm.es

Luis Rodriguez-Benitez
Universidad de Castilla-La Mancha
Escuela Superior de Informática
Ciudad Real, Spain
luis.rodriguez@uclm.es

***Abstract*—Linguistic Petri Net (LPN) is a method to generate linguistic descriptions, which maintains the way that Petri Nets (PNs) work along with the mechanisms necessary to generate linguistic descriptions of systems. This paper presents a new LPN module, called the Trends Processing Module (TrPM), to generate linguistic descriptions of trends in a time series. This takes the series as an input and returns a sequence of places that represent in detail each one of the trends of the series, which allows us to generate the description. A public dataset has been used to test the presented module. The designed module can be used to design another more complex LPNs that need an approximate model of the trends of a time series.**

***Index Terms*—Natural language description, linguistics Petri Nets, fuzzy logic, time series analysis**

I. Introduction

Brief summaries in natural language can often be more effective than traditional presentations of numerical data [1]. This has led to the growth of a relatively modern area within the field of Natural Language Generation, which is the automatic generation of linguistic descriptions of quantitative data [2]. These kinds of summaries are used, for instance, in virtual assistants such as Alexa, Google Home or SIRI.

The generation of linguistic descriptions of time series (TS) that has been acquired by all kind of sensors, technologies and observations over a time period (i.e. a TS), is called GLiDTS, which differs radically from classical techniques studying TS based on segmentation, forecasting and, pattern recognition and extraction. Marín and Sánchez [3] review the state of the literature with respect to GLiDTS and they show the relevance of the Fuzzy Sets of Zadeh, their protoforms and computation with words [4], [5]. Kacprzyk distinguishes three different generations of systems, according to the complexity and the use of Fuzzy Sets of Zadeh [6]. They highlight a third generation, which is fully linked to natural language technologies and to the concept of summaries of data collections introduced by Yager [7]. Examples of descriptions based on segmentation of series such as Kacprzyk [8], [9], J. Moreno-Garcia et al. [10] and another approach with more complex summary schemes such as Conde-Clemente sample in [11]. Meanwhile, Triviño and Sugeno [12] introduced the concept of Linguistic Granular Model of a Phenomenon (GLMP), where they combine ideas from two fields namely: the computational theory of perceptions [13] and systemic functional linguistics [14]—as a general framework for the automatic generation of descriptions in natural language, as seen in [15]. Petri Nets have been widely used to explicitly add the treatment and identification of temporal components. Moreno-Garcia et al. [16] proposed a Linguistic extension of Petri Networks, called Linguistic Petri Nets, to obtain descriptions and linguistic summaries of systems. This proposal is the backbone of this paper, where a method to linguistically identify and describe temporal situations that can be characterised by sequences of temporal data trends is shown.

It is necessary to consider the events when generating linguistic descriptions of TS. Events help to detect well-known situations such as the location of a trend, a trend change, or when a maximum or minimum is reached, etc. To deal with all these situations, it is necessary to check the input flow of TS values. PN are prepared to deal with all these situations, allowing the detection of events and the management of the flow, providing the necessary tools to synchronize and coordinate the system. The module called Trends Processing Module (TrPM) is proposed in this paper to generate linguistic descriptions of trends in a time series. A sequence of places that represents in detail each one of the trends of the series is obtained using a TS as input. These places generate the descriptions. A public dataset has been used to test the presented module. The designed module can be used to design another more complex LPNs that need an approximate model of the trends of a time series.

The rest of this paper is organised as follows: Section II briefly describes the structure and functioning of the LPNs and in Section III the module for the linguistic description of trends is defined. Section IV discusses the case of study. Our conclusions and future works are presented in Section V.

II. LPN review

This section briefly describes how the LPN operates [16], showing its structure. Formally, an LPN is a language employed to generate linguistic descriptions of a system. It is represented by a tuple $\{P, T, M\}$, where:

This work was funded by the project TIN2015-64776-C3-3-R of the Science and Innovation Ministry of Spain, co-funded by the European Regional Development Fund (ERDF), and by the Information Systems and Technologies Department of the University of Castilla-la Mancha.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

- P is a non-empty set of linguistic places $P_i = \{E_i, W_i, Alg_i\}$, where
  - E_i is an ordered set of sets of linguistic labels E_{i_j}. If E_i contains only one set, then it will be represented as a single set throughout this paper.
  - W_i is a set of sets of membership grades W_{i_j}, where each $w \in W_{i_j}$ takes values in $[0, 1]$. Each label $e \in E_{i_j}$ has a value $w \in W_{i_j}$.
  - $Alg_i = \{Tpt_i, V_i\}$ is an algorithm that generates as output a linguistic description using the template Tpt_i based on E_i, W_i, and V_i, where V_i is a set of variables that contains the relevant information taken during the evolution of the LPN.
- T is a non-empty set of processing transitions $T_i = \{I_i, O_i, l_i, c_i\}$, where:
  - I_i is the set of input places for the transition T_i;
  - O_i is the set of output places for T_i;
  - l_i is the logical function that checks if the transition must be fired and when this happens, O_i is computed using the function c_i.
  - c_i is the function needed to compute O_i when the transition is fired.
- M is a state vector, which indicates what the marking is at this instant. The initial marking in the net will be denoted by $M_0 = [m_1, m_2, \ldots, m_{|P|}]$ to distinguish it from the rest.

LPNs have places and transitions, like PNs. The transitions are equipped with the ability to fill their output places using the input places. A transition operates similarly to PN transitions: when every input place has the necessary number of marks (i.e. the transition is enabled), the logical function l is checked, and the transition is fired if that function evaluates to true. The function c then computes the components of the output places, and the state vector M is updated.

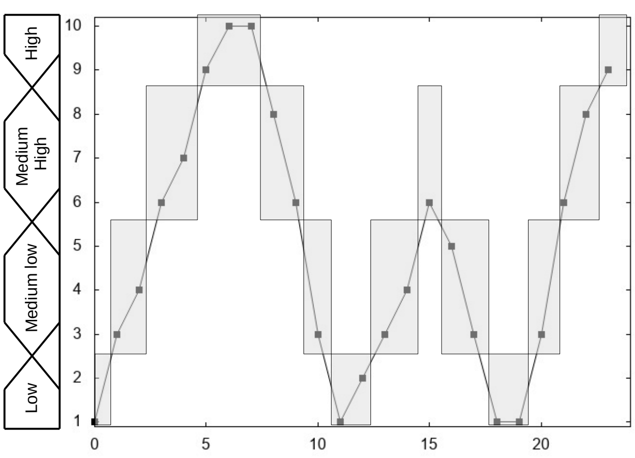

Fig. 1. Example of how the module forms groups generating the sequence of the module output places (grey rectangles). x and y axes are the time an the TS values respectively.

In this paper the module called *Time Series Processing Module (TSPM)* is used, which was first introduced in [16]. Inside LPNs, the modules can be understood as a black box that before an inlet flow offers a sequence of outlets that describe what they were designed for. The TSPM obtains as an output a place (P_x) that groups a set of consecutive samples of the TS that have their maximum membership in the same linguistic label. Figure 1 shows an example of how the module forms groups, generating the sequence of the module output places (grey rectangles). In this case TSPM produces a sequence of 14 output places. This generates descriptions similar to “The value Low holds from the instant 0 to the instant 1.6”, where Low is a linguistic label and 0 and 1.6 are the temporal instants delimiting the interval.

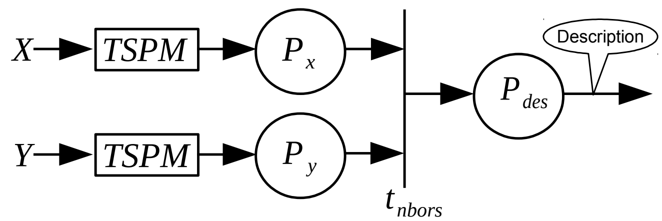

Fig. 2. Example of a LPN.

Figure 2 shows an example of a LPN that generates a linguistic description reporting the number of times that neighboring labels have been detected. It takes TS $X = \{x_1, \ldots, x_{|X|}\}$ and $Y = \{y_1, \ldots, y_{|Y|}\}$ as an input, where each x_i and y_i is the sample of the TS corresponding to the instant i. Both $TSPM$ take the TS as an input and generate the places P_x and P_y with the structure shown in Table I. P_x and P_y use the set $E_i = \{e_i^1 \ldots e_i^5\} = \{very\ low,\ low,\ middle,\ high,\ very\ high\}$ and the component Alg_i is empty. P_{des} (Table I) is the place that generates the description, which details the location area of the labels using the set of labels E_{des} and the number of times when neighbours labels have been detected using the variable $V_{des}^N \in V_{des}$.

The transition t_{nbors} is designed to store the number of times when two inputs take their neighbour's values. The transition t_{nbors} is formally represented by $t_{nbors} = \{I_{nbors}, O_{nbors}, l_{nbors},\ c_{nbors}\}$, where $I_{nbors} = \{P_x, P_y\}$, $O_{nbors} = \{P_{des}\}$ and l_{nbors} is the logical function that checks whether t_{nbors} must be fired. It is fired when the distance between the position of the *Maximum Degree Validity Label* ($MDVL$) of E_x and E_y is equal to 1 (Equation 1). $MDVL$ will be used during this document.

$$l_{nbors} = \begin{cases} True & |p_x - p_y| = 1 \\ False & in\ other\ case \end{cases} \quad (1)$$

where $p_x = MDVL(P_x) = argmax_{j \in |W_x|} w_{x_j}$ and $p_y = MDVL(P_y) = argmax_{j \in |W_y|} w_{y_j}$.

c_{nbors} gives values to the components of P_{des} that must be reassigned. These components take the following values:

- $E_{des} = \{e_i^1 \ldots e_{|E_i|}^1\} = \{left,\ central,\ right\}$.
- $W_{des} = \{w_{des_1},\ w_{des_2},\ w_{des_3}\}$ where every w_{des_i} is computed using Equation 2.

TABLE I
MAIN COMPONENTS OF THE PLACES.

$P_i = \{E_i, W_i, Alg_i\}$ taken i values x and y

Comp.	Value	Comments
E_i	$\{e_i^1 \dots e_i^5\} = \{VL, L, M, H, VL\}$	Each e_i^j is a linguistic label
W_i	$\{w_{i_1} \dots, w_{i_5}\}$	Calculated using the inputs X and Y (Figure 2)
Alg_i	$<>$	No template or variables were needed in this place

$P_{des} = \{E_{des}, W_{des}, Alg_{des}\}$

Comp.	Value	Comments
E_{des}	$\{e_{des}^1 \dots e_{des}^3\} = \{left, central, right\}$	Each e_{des}^i is a linguistic label
W_{des}	$\{w_{des_1}, w_{des_2}, w_{des_3}\}$	Calculated using the function c_{nbors}
Alg_{des}	$\{Tpt_{des}, V_{des}\}$	The algorithm and its variables
Tpt_{des}	The two system inputs are very close in the $e_i \in E_{des}$ area. This closeness has been detected V_{des}^N times.	The used template
V_{des}	$\{V_{des}^N\}$	V_{des}^N contains the number of times that neighbours has been detected

$$w_{des_i} = \mu_{e_{des_i}}(defuzzification(e_x^{p_x} \cup e_y^{p_y})) \quad (2)$$

where $i \in [1..|E_{des}|]$, $p_x = MDVL(P_x)$ and $p_y = MDVL(P_y)$.

- $V_{des}^N = V_{des}^N + 1$ because a new case of neighbour values has been detected.

The state vector is $M = [m_x, m_y, m_{des}]$ where m_i is the number of marks in P_i.

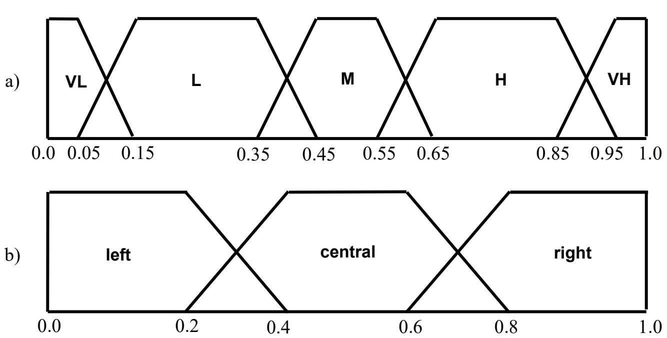

Fig. 3. a) Sets of labels used by E_x and E_y, b) Sets of labels used by E_{des}.

The operation of this net will now be described in detail to clarify the evolution of the marking. t_{nbors} will be enabled when every input place has at least one mark. Then, it will be fired if the function l_{nbors} evaluates to true. When P_x and P_y are marked ($M = [1, 1, 0]$), t_{nbors} is enabled because its two input places are marked. To explain the operation of this net, two examples are now introduced.

Example 1: Let us assume $V_{ab}^N = 2$, $P_x = \{E_x, W_x, Alg_x\} = \{ \{VL, L, M, H, VH\}, \{0, 0, 0.8, 0.2, 0\}, <>\}$ and $P_y = \{E_y, W_y, Alg_y\} = \{ \{VL, L, M, H, VH\}, \{\{0, 0, 0, 0.7, 0.3\}, <>\}$. Transition t_{nbors} is fired because $p_x = MDVL(P_x) = 3$ and $p_y = MDVL(P_y) = 4$ and it is satisfied $l_{nbors} = |p_x - p_y| = |3 - 4| = 1$ (Equation 1). c_{nbors} assigns the values of P_{des} as follows:

- $E_{des} = \{left, central, right\}$.
- $W_{des} = \{0, 0.75, 0.25\}$ because $e_x^{p_x} = e_x^3 = M$ and $e_x^{p_y} = e_y^4 = H$, then $defuzzification(M \cup H) = defuzzification([M, H]) = 0.65$ (Figure 3, the Mean of Maxima Method (MOM) method [17] has been used) and then $\mu_{left}(0.65) = 0.0$, $\mu_{central}(0.65) = 0.75$ and $\mu_{right}(0.65) = 0.25$. The label *central* is selected to localize the area since it has the maximum membership grade.
- Alg_{des} uses the template shown in Table I and $V_{des}^N = V_{des}^N + 1 = 2 + 1 = 3$.

The following description is then generated: "The two system inputs are very close in the *central* area. This closeness has been detected *3* times."

Example 2: Let us assume $P_x = \{E_x, W_x, Alg_x\} = \{ \{VL, L, M, H, VH\}, \{0, 0.8, 0.2, 0, 0\}, <>\}$ and $P_y = \{E_y, W_y, Alg_y\} = \{\{VL, L, M, H, VH\}, \{0, 0, 0, 0.7, 0.3\}, <>\}$. Transition t_{nbors} is not fired because $p_x = 2$ and $p_y = 4$, and then $l_{des} = |p_x - p_y| = |2 - 4| = 2 \neq 1$ (Equation 1). As in PN, the designed LPN must contain another transition that in these situations must be fired, allowing the net to evolve.

Once briefly explained, the concept and operation of LPN next section introduces the new module related to trends in TS.

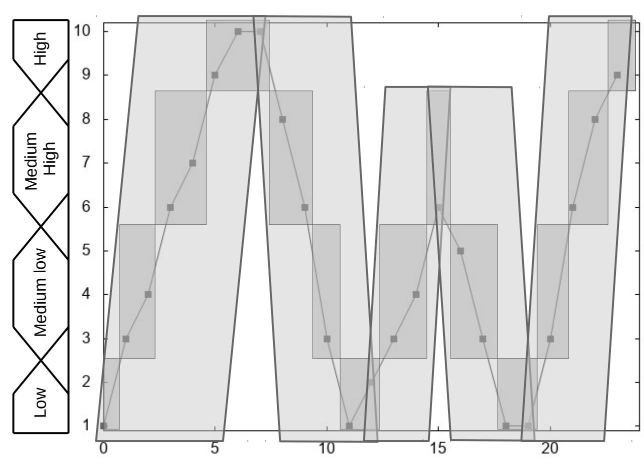

Fig. 4. An example of how the Trend Processing Module groups consecutive output places to create trends. Each parallelogram represents a trend.

III. TRENDS PROCESSING MODULE

The study of trends in TS usually classifies them into *flat*, *increasing* and *decreasing*. Moreno et. al [16] introduced a LPN to detect *increasing* and *decreasing* trends. This section presents a new module to detect these three types. This module

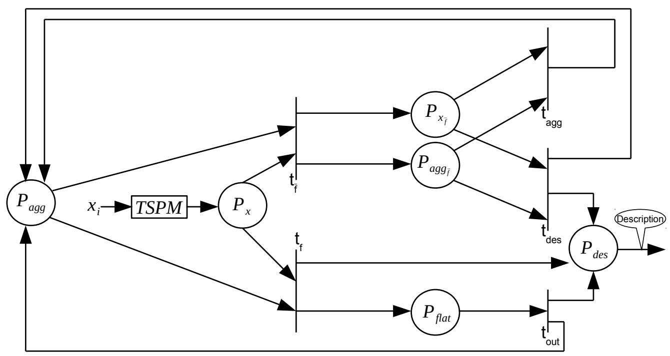

Fig. 5. Trend Processing Module (TrPM).

takes as an input the TS, generating a place that models every detected trend. The information on the module's output place is considerably more detailed, allowing much more complete descriptions to be generated. Figure 4 shows an example of how $TrPM$ groups consecutive output places to create trends. In this case the module generates four trends (parallelogram). Figure 5 shows the design of the TrPM where the TSPM is included. The TrPM will now be formally detailed, which is composed of six places and five transitions. The unique input to the network is the TS $X = \{x_1, x_2, \dots, x_{|X|}\}$ where each x_i corresponds to the place of the TS in the instant i, whilst $|X|$ is the number of elements of X. Every x_i is processed by the module $TSPM$ [16] and the place P_x is obtained. P_x and P_{agg} are the inputs to t_f and $t_{\overline{f}}$. They detect whether or not a *flat* trend is present. These transitions are mutually exclusive, so only one of them can be fired; that is, the control flow only continues through one of them. When a *flat* trend is detected, t_f must be fired to generate the two linguistics descriptions of the previous trend (P_{agg}) and the own *flat* trend (P_x). With this aim, t_f copies P_{agg} to P_{des} (it generates the description) and P_{flat} is created. Finally, t_{out} will copy P_{flat} to P_{des} to generate the description of the *flat* trend.

If no *flat* trend is detected, then $t_{\overline{f}}$ is fired copying P_x and P_{agg} to $P_{x_{\overline{f}}}$ and $P_{agg_{\overline{f}}}$, respectively. Basically, the process consists of an aggregation in P_{agg} of all the consecutive places of the same trend until a change is detected. Both situations are managed by means of t_{agg} and t_{des} (mutually exclusive). Each of them are responsible of:

- t_{agg} is fired if the same trend continues, generating a new P_{agg} from $P_{agg_{\overline{f}}}$ and $P_{x_{\overline{f}}}$.
- t_{des} is fired if a change in the trend is detected and copies $P_{agg_{\overline{f}}}$ to P_{des} generating its description. Furthermore, it creates a new P_{agg} from $P_{agg_{\overline{f}}}$ and $P_{x_{\overline{f}}}$.

This module offers a set of consecutive places as an output, each representing a trend.

Of all the places in the network, the most outstanding are P_x and P_{agg} because the others are copies of them. Copies are identified using the sub-index of the transition that completes such copy. For instance, $P_{x_{\overline{f}}}$ is a replica of P_x completed by $t_{\overline{f}}$, and $P_{agg_{\overline{f}}}$ is the copy of P_{agg} carried out by t_f. P_{des} is the place that generates the description and it is a copy of

TABLE II
MAIN COMPONENTS OF P_x AND P_{agg}.

$P_x = \{E_x, W_x, Alg_x\}$

Comp.	**Value**
E_x	$\{e_x^1, \dots, e_x^{\lvert E_x \rvert}\}$
W_x	$\{w_{x1}, w_{x2}, \dots, w_{x_{\lvert E_x \rvert}}\}$
Alg_x	$\{Tpt_x, V_x\}$
Tpt_x	<>
V_x	$\{V_x^{ini}, V_x^{fin}\}$

$P_{agg} = \{E_{agg}, W_{agg}, Alg_{agg}\}$

Comp.	**Value**
E_{agg}	$\{-e_{agg}^{\lvert E_{agg} \rvert}, \dots, -e_{agg}^1, e_{agg}^1, \dots, e_{agg}^{\lvert E_{agg} \rvert}\}$
W_{agg}	$\{w_{agg_{-\lvert E_{agg} \rvert}}, \dots, w_{agg_{-1}}, w_{agg_1}, \dots, w_{agg_{\lvert E_{agg} \rvert}}\}$ where W_{agg} is governed by Equation 3 and it is assigned by t_f or $t_{\overline{f}}$. $w_{agg_i} = \begin{cases} 1 & ((p_{agg} \geq 1) \wedge (p_{agg} \leq i \leq p_x)) \\ & \vee ((p_{agg} \leq -1) \wedge (p_{agg} \leq i \leq -p_x)) \\ 0 & in\ other\ case \end{cases}$ (3) where $p_{agg} = \{j \mid (w_{agg_j} = 1)\ and\ (\forall k < j)\ (w_{agg_k} < 1)\}$ and $p_x = MDVL(P_x)$.
Alg_{agg}	$\{Tpt_{agg}, V_{agg}\}$
Tpt_{agg}	There is a V_{agg}^{type} trend from $V_{agg}^{L_{ini}}$ to $V_{agg}^{L_{fin}}$ during the interval V_{agg}^{ini} to V_{agg}^{fin}
V_{agg}	$\{V_{agg}^{ini}, V_{agg}^{fin}, V_{agg}^{L_{ini}}, V_{agg}^{L_{fin}}, V_{agg}^{type}, V_{agg}^{list} = \{v_{agg}^{list^1}, \dots, v_{agg}^{list^{\lvert V_{agg}^{list} \rvert}}\}\}$ where $v_{agg}^{list^i} = [label_{agg}^{list^i}, ini_{agg}^{list^i}, fin_{agg}^{list^i}]$ being $i \in [1, .., \lvert V_{agg}^{list} \rvert]$

$P_{agg_{\overline{f}}}$ when t_{des} is fired, or the replica of P_{agg} or P_{flat} if t_f or t_{out} are fired, respectively.

Table II summarises the most relevant elements of P_x y P_{agg}:

- P_x: These components are returned by the TSPM [16]. Alg_x does not use the template but their variables $V_x = \{V_x^{ini}, V_x^{fin}\}$ to indicate the initial and final instants of the represented trend.
- P_{agg}: Their components are:
  - $E_{agg} = \{-e_{agg}^{\lvert E_{agg} \rvert}, \dots, -e_{agg}^1, e_{agg}^1, \dots, e_{agg}^{\lvert E_{agg} \rvert}\}$: this is its set of labels and E_x is needed to obtain it. First, $\dot{E}_x$ is used because it contains the "reverse label" for each label $E_x^i \in E_x$. The "reverse label" of $e_x^i = \{a, b, c, d\}$ (trapezoidal) is $e_x^{-i} = \{-d, -c, -b, -a\}$. Then, for instance if $e_x^i = \{0.5, 2, 4, 4.5\}$ then $e_x^{-i} = \{-4.5, -4, -2, -0.5\}$. $\dot{E}_x$ is needed to represent decreasing trends. E_{agg} is the result of the union of $\dot{E}_x$ and E_x ($E_{agg} = \dot{E}_x \cup E_x$); that is, $E_{agg} = \{e_x^{-n}, \dots, e_x^{-1}, e_x^1, \dots, e_x^n\}$. The order criterion for the indexes is $-n < \cdots < -1 < 1 < \cdots < n$.
  - $W_{agg} = \{w_{x_{-n}}, \dots, w_{x_{-1}}, w_{x_1}, \dots, w_{x_n}\}$: Each w_{x_i} with $i \in [-|E_{agg}|, \dots, -1, 1, \dots, |E_{agg}|]$ indicates when e_x^i is part of the trend (Equation 3). With this aim, all of the membership values of the labels that are part of the trend modelled by P_{agg}

TABLE III
VALUES TAKEN BY t_f AND t_{out}.

$t_f = \{I_f, O_f, l_f, c_f\}$

Comp.	Value
I_f	$\{P_x, P_{agg}\}$
O_f	$\{P_{des}, P_{flat}\}$
l_f	$$l_f = \begin{cases} True & \lvert V_x^{fin} - V_x^{ini} + 1\rvert > time_f^{min} \\ False & in\ other\ case \end{cases} \quad (4)$$
c_f	copies P_{agg} to P_{des} and calculates P_{flat}

$t_{out} = \{I_{out}, O_{out}, l_{out}, c_{out}\}$

Comp.	Value
I_{out}	$\{P_{flat}\}$
O_{out}	$\{P_{des}\}$
l_{out}	no condition
c_{out}	copies P_{flat} to P_{des} and generates P_{agg} using P_{flat}

take the value 1.

– $Alg_{agg} = \{Tpt_{agg}, V_{agg}\}$: The template is "*There is a V_{agg}^{type} trend from $V_{agg}^{L_{ini}}$ to $V_{agg}^{L_{fin}}$ during the interval V_{agg}^{ini} to V_{agg}^{fin}.*". This is why V_{agg} contains $\{V_{agg}^{ini}, V_{agg}^{fin}, V_{agg}^{L_{ini}}, V_{agg}^{L_{fin}}, V_{agg}^{type}, V_{agg}^{list}\}$ where V_{agg}^{ini} and V_{agg}^{fin} are the initial and final instants that represents the trend; $V_{agg}^{L_{ini}}$ and $V_{agg}^{L_{fin}}$ are the first and last labels of the trend; V_{agg}^{type} indicates the kind of trend: *increasing*, *decreasing* or $flat$; and finally, V_{agg}^{list} is a list of tuples $v_{agg}^{list^i} = [label_{agg}^{list^i}, ini_{agg}^{list^i}, fin_{agg}^{list^i}]$ with $i \in [1, .., |V_{agg}^{list}|]$. Each element $label_{agg}^{list^i}$ contains the label, and the initial and final instants of each place P_x added to P_{agg}. An example of how this template can be used is: "There is a *decreasing* trend from *high* to *low* during the interval 0.23 to 0.62".

The transitions will now be detailed. First, the part responsible of detecting a trend of type *flat* (t_f) is studied, and then the rest of the cases are studied ($t_{\overline{f}}$). To detect and process a *flat* trend, t_f and t_{out} are used (Table III). t_f transfers P_{agg} to P_{des} computing P_{flat} by means of P_{agg} and P_x. P_{flat} has a similar structure than P_{agg} (Table II):

- E_{flat} is copied from P_{agg}.
- W_{flat} is filled by t_f using Equation 3 (Table II).
- Alg_{flat}: Tpt_{flat} is similar to Tpt_{agg}. V_{flat} has the same set of variables than V_{agg}, formally $V_{flat} = \{V_{flat}^{ini}, V_{flat}^{fin}, V_{flat}^{L_{ini}}, V_{flat}^{L_{fin}}, V_{flat}^{type}, V_{flat}^{list}\}$ being $V_{flat}^{ini} = V_x^{ini}$, $V_{flat}^{fin} = V_x^{fin}$, $V_{flat}^{L_{ini}} = e_x^{p_x}$ and $V_{flat}^{L_{fin}} = e_x^{p_x}$ where $p_x = MDVL(P_x)$ and $V_{flat}^{L_{type}} = flat$. V_{flat}^{list} is equal to $[[label, ini, fin]]$, $label = e_{agg}^{p_x}$ with $p_x = MDVL(P_x)$, $ini = V_x^{ini}$ and $fin = V_x^{fin}$. All variables take their corresponding values from P_x with the exception of $V_{flat}^{L_{type}}$, which is assigned to *flat*.

t_f is activated once l_f is satisfied; that is, when the duration of P_x is greater than the threshold value ($time_f^{min}$). This value is established during the creation of the network (Equation 4 in Table III).

Then, it is activated t_{out} and transfers P_{flat} to P_{des}, and it generates P_{agg} from P_{flat} take the next values:

- E_{agg} is copied from P_{flat}.
- W_{agg} is obtained from E_{flat}.
- Alg_{agg}: Tpt_{agg} is equal to Tpt_{flat} and $V_{agg} =$

TABLE IV
VALUES TAKEN BY $t_{\overline{f}}$, t_{agg} AND t_{des}.

$t_{\overline{f}} = \{I_{\overline{f}}, O_{\overline{f}}, l_{\overline{f}}, c_{\overline{f}}\}$

Comp.	Value
$I_{\overline{f}}$	$\{P_x, P_{agg}\}$
$O_{\overline{f}}$	$\{P_{x_{\overline{f}}}, P_{agg_{\overline{f}}}\}$
$l_{\overline{f}}$	*not* l_f
$c_{\overline{f}}$	copies P_x and P_{agg} to $P_{x_{\overline{f}}}$ and P_{agg}, respectively.

$t_{agg} = \{I_{agg}, O_{agg}, l_{agg}, c_{agg}\}$

Comp.	Value
I_{agg}	$\{P_{x_{\overline{f}}}, P_{agg_{\overline{f}}}\}$
O_{agg}	$\{P_{agg}\}$
l_{agg}	$$l_{agg} = \begin{cases} True & ((p_{agg} \geq 1)\ \wedge\ (p_{agg} < p_x))\ \vee \\ & ((p_{agg} \leq -1)\ \wedge\ (\lvert p_{agg}\rvert > p_x)) \\ False & in\ other\ case \end{cases} \quad (5)$$ where $p_{agg} = \{j \mid (w_{agg_j} = 1)\ and\ (\forall i > j)\ (w_{agg_i} < 1)\}$ and $p_x = MDVL(P_x)$.
c_{agg}	It creates a new P_{agg} from $P_{x_{\overline{f}}}$ to $P_{agg_{\overline{f}}}$.

$t_{des} = \{I_{des}, O_{des}, l_{des}, c_{des}\}$

Comp.	Value
I_{des}	$\{P_{x_{\overline{f}}}, P_{agg_{\overline{f}}}\}$
O_{des}	$\{P_{des}, P_{agg}\}$
l_{des}	*not* l_{agg}
c_{des}	It copies $P_{agg_{\overline{f}}}$ to P_{des} and creates P_{agg} from $P_{x_{\overline{f}}}$ and $P_{agg_{\overline{f}}}$.

TABLE V
EXAMPLES OF HOW TO DETECT IF A TREND REMAINS, WITH $E_x = \{VL, L, M, H, VH\}$ AND $E_{agg} = \{-VH, -H, -M, -L, -VL, VL, L, M, H, VH\}$.

Values	Explanation
$W_{agg} = \{0, 0, 0, 0, 0, 0, 0, 1, 1, 0\}$: increasing trend from M to H $W_x = \{0, 0, 0.0, 0.25, 0.75\}$ (label VL)	$p_{agg} = 4$ and $p_x = 5$, then Equation 5 yields that $(p_{agg} \geq 1)$ and $(p_{agg} < p_x) = (4 < 5)$, then l_{agg} returns $true$ and t_{agg} is fired. The aggregation of P_{agg} and P_x is M, H, VH.
$W_{agg} = \{0, 1, 1, 0, 0, 0, 0, 0, 0, 0\}$: decreasing trend from H to M $W_x = \{0.6, 0.4, 0, 0, 0\}$ (label VL)	$p_{agg} = -4$ and $p_x = 1$, then Equation 5 (Table IV) yields that $(p_{agg} \leq -1)$ and $(\lvert p_{agg}\rvert > p_x) = (4 > 1)$, then l_{agg} returns $true$ and t_{agg} is fired. The aggregation of P_{agg} and P_x is H, M, L, VL.

$\{V_{agg}^{ini}, V_{agg}^{fin}, V_{agg}^{L_{ini}}, V_{agg}^{L_{fin}}, V_{agg}^{type}, V_{agg}^{list}\}$ being $V_{agg}^{ini} = V_{flat}^{fin}$, $V_{agg}^{fin} = V_{flat}^{fin}$, $V_{agg}^{L_{ini}} = e_{flat}^{p}$ and $V_{agg}^{L_{fin}} = e_{flat}^{p}$ where $p = MDVL(P_{flat})$ and $V_{flat}^{L_{type}} = flat$. V_{agg}^{list} is equal to $[[label, ini, fin]]$, $label = e_{agg}^{p}$, $ini = V_{flat}^{fin}$ and $fin = V_{flat}^{fin}$.

For those situations where no *flat* trend is detected, the transitions shown in Table IV are needed. Transition $t_{\overline{f}}$ is fired when P_x does not represent a *flat* trend, transferring P_x and P_{agg} to $P_{x_{\overline{f}}}$ and $P_{agg_{\overline{f}}}$, inputs of t_{agg} and t_{des}. Then, the flow control of the network is passed to one of the mutually exclusive transitions t_{agg} and t_{des}. Formally, the components of t_{agg} are:

- $I_{agg} = \{P_{agg_{\overline{f}}}, P_{x_{\overline{f}}}\}$.
- $O_{agg} = \{P_{agg}\}$.
- l_{agg}: Equation 5 from Table IV detects when an increasing trend, $p_{agg} \geq 1$, or a decreasing trend ($p_{agg} \leq -1$) remains. Table V shows an example for each one of these possible situations.
- c_{agg}: creates the new P_{agg} from the two inputs of t_{agg} taking their components next values:
  - E_{agg} is a copy of $E_{agg_{\overline{f}}}$.
  - W_{agg} is governed by the Equation 3 (Table II). An example of how it works is shown in Table VI.
  - Alg_{agg}: Tpt_{agg} is equal to $Tpt_{agg_{\overline{f}}}$ and the components of V_{agg} take the next values $V_{agg}^{ini} = V_{agg_{\overline{f}}}^{ini}$, $V_{agg}^{fin} = V_{x_{\overline{f}}}^{fin}$, $V_{agg}^{L_{ini}} = V_{agg_{\overline{f}}}^{L_{ini}}$ and $V_{agg}^{L_{fin}} = e_{agg}^{p_x}$ with $p_x = MDVL(P_{x_{\overline{f}}})$ and $V_{agg}^{L_{type}}$ is assigned according to Equation 6.

$$V_{agg}^{L_{type}} = \begin{cases} increasing & |V_{x_{\overline{agg}}}^{ini}| < |V_{x_{\overline{f}}}^{fin}| \\ decreasing & in\ other\ case \end{cases} \quad (6)$$

V_{agg}^{list} is equal to $append(V_{agg_{\overline{f}}}^{list}, v)$ being $v = [label, ini, fin]$, $label = e_{agg}^{p_x}$, $ini = V_{x_{\overline{f}}}^{ini}$ and $fin = V_{x_{\overline{f}}}^{fin}$.

Formally, $t_{des} = \{I_{des}, O_{des}, l_{des}, c_{des}\}$ where:

- $I_{des} = \{P_{agg_{\overline{f}}}, P_{x_{\overline{f}}}\}$.
- $O_{des} = \{P_{agg}, P_{des}\}$.
- l_{des} is the negation of l_{agg} (Equation 5).
- c_{des} transfers $P_{agg_{\overline{agg}}}$ to P_{des}. Furthermore, c_{des} creates P_{agg} from $P_{agg_{\overline{agg}}}$ and $P_{x_{\overline{f}}}$ being their components:
  - E_{agg} is copied from $E_{agg_{\overline{agg}}}$.
  - W_{agg} is filled using the last label from $E_{agg_{\overline{f}}}$ and the unique label from $P_{x_{\overline{f}}}$. For instance, let $W_{agg_{\overline{f}}} = \{0,0,0,0,0,1,1,1,1,0\}$ (increasing trend from VL to H) and $W_{x_{\overline{f}}} = \{0,0,1,0,0\}$ (representing label M), which is detected the end of the trend in $P_{agg_{\overline{f}}}$ and the new decreasing trend is $[H, M]$; that is, $W_{agg} = \{0,1,1,0,0,0,0,0,0,0\}$. For example, let $W_{agg_{\overline{f}}} = \{0,0,1,1,1,0,0,0,0,0\}$ ($[M, L, VL]$) and $W_{x_{\overline{f}}} = \{0,1,0,0,0\}$ (L), then the new trend is $[VL, L]$; that is, $W_{agg} = \{0,0,0,0,0,0,1,1,0,0\}$. Equation 7 governs how W_{agg} is assigned. The first case processes the change from an increasing trend ($p_{agg_{\overline{f}}} \geq 0$) to a decreasing trend, and the second the opposite trend ($p_{agg_{\overline{f}}} < 0$).

$$w_{agg_i} = \begin{cases} 1 & (p_{agg_{\overline{f}}} \geq 0)\ and\ (-p_{agg_{\overline{f}}} \leq i \leq -p_x) \\ 1 & (p_{agg_{\overline{f}}} < 0)\ and\ (|p_{agg_{\overline{f}}}| \leq i \leq p_x) \\ 0 & in\ other\ case \end{cases} \quad (7)$$

where $p_{agg_{\overline{f}}} = \{j \mid (w_{agg_{\overline{f}}}^{j} = 1)\ and\ (\forall k > j)\ (w_{agg_{\overline{f}}}^{k} < 1)\}$ and $p_x = MDVL(P_{x_{\overline{f}}})$.

  - V_{agg} takes the values: $V_{agg}^{ini} = V_{agg_{\overline{f}}}^{fin}$ y $V_{agg}^{fin} = V_{x_{\overline{f}}}^{fin}$, $V_{agg}^{L_{ini}} = V_{agg_{\overline{f}}}^{fin}$ and $V_{agg}^{L_{fin}} = V_{x_{\overline{f}}}^{ini}$ and $V_{agg}^{L_{type}}$ is assigned according to Equation 8.

$$w_{agg_i} = \begin{cases} increasing & isDecreasing(V_{agg_{\overline{f}}}^{L_{type}}) \\ decreasing & in\ other\ case \end{cases} \quad (8)$$

V_{agg}^{list} is equal to $append(V_{agg_{\overline{f}}}^{list}, v)$ being $v = [label, ini, fin]$, $label = e_{agg}^{p_x}$ with $p_x = MDVL(P_{x_{\overline{f}}})$, $ini = V_{x_{\overline{f}}}^{ini}$ and $fin = V_{x_{\overline{f}}}^{fin}$.

Table VII shows the evolution of P_{agg}, while a trend is aggregated when a sequence of consecutive $P_{x_{\overline{f}}}$ appears. The steps to follow are:

TABLE VI
EXAMPLES OF HOW c_{agg} CALCULATES THE NEW W_{agg}.

Values	Explanation
$W_{agg} = \{0,0,0,0,0,0,0,1,1,0\}$: increasing trend from M to H $W_x = \{0,0,0.0,0.25,0.75\}$ (label VH)	$p_{agg} = 3$ and $p_x = 5$, then the first case of Equation 3 is used since $p_{agg} \geq 1$. The aggregation is done assigning the new $W_{agg} = \{0,0,0,0,0,0,0,1,1,1\}$; that is, w_{x_i} is assigned to 1 if $(p_{agg} \leq i \leq p_x)$, in this case, $(3 \leq i \leq 5)$, and the new trend is M, H, VH.
Values	**Explanation**
$W_{agg} = \{0,1,1,0,0,0,0,0,0,0\}$: decreasing trend from M to H $W_x = \{0.6,0.4,0,0,0\}$ (label VL)	$p_{agg} = -4$ and $p_x = 1$ the second case of Equation 3 is used since $(p_{agg} \leq -1)$; that is, $(-4 \leq -1)$. The aggregation is done assigning the new $W_{agg} = \{0,1,1,1,1,0,0,0,0,0\}$; that is, w_{x_i} is assigned to 1 if $(p_{agg} \leq i \leq -p_x)$, in this case, $(-4 \leq i \leq -1)$, and the new trend is H, M, L, VL.

TABLE VII
EVOLUTION OF V_{agg} WHEN A SEQUENCE OF $V_{x_{\overline{f}}}$ APPEARS.

$V_{x_{\overline{f}}}$	$V_{agg_{\overline{f}}} = \{V_{agg}^{ini}, V_{agg}^{fin}, V_{agg}^{L_{ini}}, V_{agg}^{L_{fin}}, V_{agg}^{type}, V_{agg}^{list}\}$
	$\{7.0, 8.0, VH, H, decreasing,$ $\{\{VH, 7.0, 7.0\}, \{H, 7.0, 8.0\}\}\}$
$\{M, 8.0, 9.0\}$	$\{7.0, 9.0, VH, M, decreasing,$ $\{\{VH, 7.0, 7.0\}, \{H, 7.0, 8.0\}, \{M, 8.0, 9.0\}\}\}$
$\{L, 9.0, 10.0\}$	$\{7.0, 10.0, VH, L, decreasing,$ $\{\{VH, 7.0, 7.0\}, \{H, 7.0, 8.0\}, \{M, 8.0, 9.0\},$ $\{L, 9.0, 10.0\}\}\}$

TABLE VIII
OUTPUT PLACES SEQUENCE OF THE TSPM MODULE WHEN THE SHUTTLE DATASET IS USED AS INPUT. EACH PLACE $P_{x_i} = \{Label_i, T_{i_1}, T_{i_2}\}$ CONTAINS THE OUTPUT LABEL NAME ($Label_i$) AND THE INITIAL AND FINAL INSTANTS (T_{i_1} AND T_{i_2}). THE LABELS $Very\ Low$, Low, $Medium$, $High$ AND $Very\ high$ ARE AS VL, L, M, H AND VH RESPECTIVELY.

$P_{x_1} = \{H, 0, 41\}$	$P_{x_{10}} = \{M, 552, 566\}$
$P_{x_2} = \{M, 41, 56\}$	$P_{x_{11}} = \{L, 566, 702\}$
$P_{x_3} = \{H, 56, 183\}$	$P_{x_{12}} = \{VL, 702, 711\}$
$P_{x_4} = \{VH, 183, 273\}$	$P_{x_{13}} = \{L, 711, 718\}$
$P_{x_5} = \{H, 273, 410\}$	$P_{x_{14}} = \{M, 718, 810\}$
$P_{x_6} = \{VH, 410, 486\}$	$P_{x_{15}} = \{H, 810, 813\}$
$P_{x_7} = \{H, 486, 528\}$	$P_{x_{16}} = \{M, 813, 828\}$
$P_{x_8} = \{M, 528, 534\}$	$P_{x_{17}} = \{L, 828, 999\}$
$P_{x_9} = \{L, 534, 552\}$	

1) Let the first $V_{agg} = \{7.0, 8.0, VH, H, decreasing, \{\{VH, 7.0, 7.0\}, \{H, 7.0, 8.0\}\}\}$ as consequence of the detection of a trend change by t_{des}.
2) Let the next place generated $V_{x_{\overline{f}}} = \{M, 8.0, 9.0\}$, then V_{agg} takes the value shown in the second column of the third row in Table VII. As can be observed, V_{agg}^{fin} and $V_{agg}^{L_{fin}}$ are assigned to $V_{x_{\overline{f}}}^{fin}$ and $V_{x_{\overline{f}}}^{fin}$ respectively, V_{agg}^{type} is updated and P_x is concatenated to V_{agg}^{list}.
3) Next, let us suppose $V_{x_{\overline{f}}} = \{L, 9.0, 10.0\}$, and the same process as the one indicated in the previous step is completed.
4) This process finishes when t_{des} is fired. This means that a trend change has been detected, transferring $P_{agg_{\overline{f}}}$ to P_{des} to generate its description.

As can be seen, P_{des} can generate a detailed description because it contains all of the trend information, both general trend information and information on the different components of the trend, stored in V_{agg}^{list}. The template introduced in Table II can be modified to contain much more detail.

IV. A CASE OF STUDY

An example of the method operation in a time series called "*shuttle*" is shown. This public dataset was used in [18]. The parameters of TSPM and TrPM have been set to $\alpha = \frac{1}{3}$ (this parameter indicates the weight that is given to the new values that enter in the TSPM with respect to those that are already represented), and the minimum length of the flat segments has been determined at 12% of the total length of the *shuttle* time series. In addition, the set of linguistic labels shown in Figure 3.a has been used in the test.

The designed module makes use of the TSPM module. Table 6 shows the sequence of output places obtained for this module. It can be seen how this sequence represents the temporality of the input series using consecutive labels between the different consecutive output places. In the same way the time is represented by consecutive time intervals for consecutive places. Figure 6 shows the label and its time interval used to represent each zone of the time series.

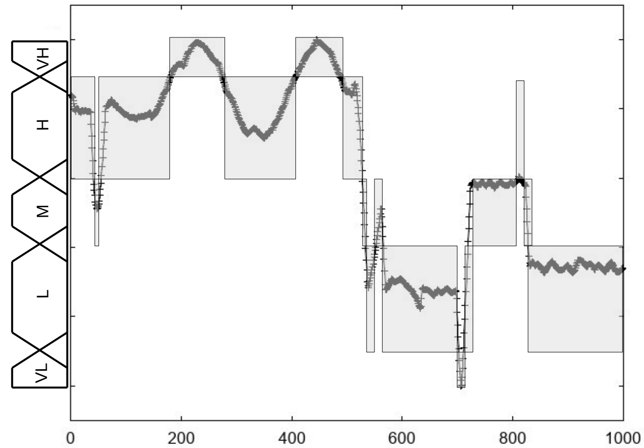

Fig. 6. Example of how TSPN forms groups using Shuttle dataset.

TABLE IX
SEQUENCE OF TRPM OUTPUT PLACES.

$P_{des_1} = \{0, 56, \{H, M\}, DEC\}$
$P_{des_2} = \{56, 183, \{H\}, FLAT\}$
$P_{des_3} = \{183, 273, \{H, VH\}, INC\}$
$P_{des_4} = \{273, 410, \{H\}, FLAT\}$
$P_{des_5} = \{410, 486, \{H, VH\}, INC\}$
$P_{des_6} = \{410, 552, \{VH, H, M, L, \}, DEC\}$
$P_{des_7} = \{534, 566, \{L, M\}, INC\}$
$P_{des_8} = \{566, 702, \{L\}, FLAT\}$
$P_{des_9} = \{702, 711, \{L, VL\}, DEC\}$
$P_{des_{10}} = \{702, 813, \{VL, L, M, H\}, INC\}$
$P_{des_{11}} = \{813, 828, \{H, M, \}, DEC\}$
$P_{des_{12}} = \{828, 999, \{L\}, FLAT\}$

The TrPM module generates its sequence of output places using as input the TSPM output places. Table IX shows the sequence of TrPM output places. This table shows how the consecutive labels that represent a trend are grouped. Figure 7 shows the output of this module. It can be seen how the location area of the trends is motivated by the design of the set of labels. For example, the second trend *flat* is due to the broad support of the *high* label, and the last detected peak could has been represented as a *flat* label, but the design of the labels causes the detection of a peak.

Finally, the descriptions obtained by the three first places generated by the TrPM module are shown:

1) There is a *decreasing* trend from *High* to *Medium* during the interval from 0 to 56.
2) There is a *flat* trend from *High* to *High* during the interval from 56 to 183.
3) There is a *increasing* trend from *High* to *Very High* during the interval from 183 to 273.

It can be seen, the linguistic descriptions offer information about the trends, more concretely, their time interval, their location area and their type.

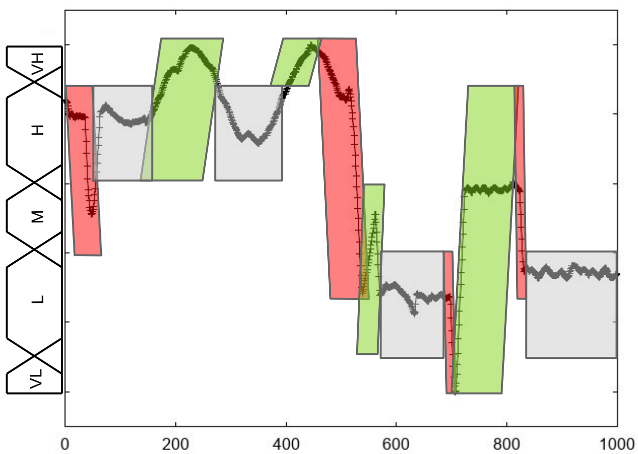

Fig. 7. Example of how TrPM form the trends using Shuttle dataset. The colors red, green and grey are used to represent the decremental, incremental and flat trends.

V. Conclusions and future works

A new method for generating linguistic descriptions of the trends of a TS has been presented in this document. This method makes use of the LPN. More specifically a new LPN module, called Trends Processing Module, has been designed to obtain in detail each one of the trends of the series, allowing us to generate the descriptions of the same one. A public dataset has been used to test the presented module. This module can be used to design another LPN.

As future work we want to design a LPN to process the TS of the electricity consumption of some buildings of University of Castilla-La Mancha to detect the relevant states of the daily cycle of the electrical consumption using the sequence of TrPM output places. Also we plan to design new LPN that make use of this module to analyze the trends in economic time series, sport, health sciences, and so on. In addition, and depending on these applications, the linguistic descriptions to be generated may have a higher level of complexity, so it will be necessary to study their impact on the module presented here.

References

[1] L. Ferres, A. Parush, S. Roberts, G. Lindgaard, "Helping people with visualimpairments gain access to graphical information through natural language: The igraph system", in: International Conference on Computers for Handicapped Persons, Springer, pp. 1122–1130, 2006.

[2] N. Marín, D. Sánchez, "Fuzzy sets and systems + natural language generation: A step forward in the linguistic description of time series", Fuzzy Sets and Systems, vol. 285, pp. 1–5, 2016.

[3] N. Marín, D. Sánchez, "On generating linguistic descriptions of time series", Fuzzy Sets and Systems, vol. 285, pp. 6–30, 2016.

[4] L. A. Zadeh, "A prototype-centered approach to adding deduction capability to search engines-the concept of protoform", in: 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings, IEEE, 2002, pp. 523–525, 2002.

[5] J. Kacprzyk, S. Zadrożny, "Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools", Information Sciences 173 (4) (2005) 281–304, 2005.

[6] J. Kacprzyk, R. R. Yager, J. M. Merigó, "Towards human-centric aggregation via ordered weighted aggregation operators and linguistic data summaries: A new perspective on zadeh's inspirations", IEEE Comp. Int. Mag., vol. 14 (1), 16–30, 2019.

[7] R. R. Yager, "A new approach to the summarization of data", Information Sciences, vol. 28 (1), pp. 69–86 (1982).

[8] J. Kacprzyk, A. Wilbik, S. Zadrożny, "Linguistic summarization of time series using a fuzzy quantifier driven aggregation", Fuzzy Sets and Systems, vol. 159 (12), pp. 1485–1499, 2008.

[9] J. Kacprzyk, S. Zadrozny, "Fuzzy logic-based linguistic summaries of time series: a powerful tool for discovering knowledge on time varying processes and systems under imprecision", Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 6 (1), pp. 37–46, 2016.

[10] J. Moreno-Garcia, L. Rodriguez-Benitez, J. Giralt, E. Del Castillo, "The generation of qualitative descriptions of multivariate time series using fuzzy logic", Applied Soft Computing, vol. 23, pp. 546–555, 2014.

[11] P. Conde-Clemente, J. M. Alonso, G. Triviño, "Toward automatic generation of linguistic advice for saving energy at home", Soft Comput. , vol. 22 (2), pp. 345–359, 2018.

[12] G. Trivino, M. Sugeno, "Towards linguistic descriptions of phenomena", International Journal of Approximate Reasoning, vol. 54 (1), pp. 22–34, 2013.

[13] L. A. Zadeh, "From computing with numbers to computing with words. from manipulation of measurements to manipulation of perceptions", IEEE Transactions on circuits and systems I: fundamental theory and applications, vol. 46 (1), pp. 105–119, 1999.

[14] M. A. K. Halliday, C. Matthiessen, M. Halliday, An introduction to functional grammar, Routledge, 2014.

[15] P. Conde-Clemente, G. Trivino, J. M. Alonso, "Generating automatic linguistic descriptions with big data", Information Sciences, vol. 380, pp. 12–30, 2017.

[16] J. Moreno-Garcia, L. Rodriguez-Benitez, L. Jimenez-Linares, G. Trivino, "A linguistic extension of petri nets for the description of systems: An application to time series", IEEE Transactions on Fuzzy Systems, vol. 27(9), pp. 1818 - 1832, 2019.

[17] W. V. Leekwijck, E. E. Kerre, "Defuzzification: criteria and classification", Fuzzy Sets and Systems, vol. 108 (2), pp. 159 – 178, 1999.

[18] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick, "Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations", IEEE Pattern Analysis and Machine Intelligence, vol. 32(12), pp. 2232–2245, 2010.