Contemporary Engineering Sciences, Vol. 11, 2018, no. 17, 793 - 806
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ces.2018.8239

An Approach for the Structural Learning of Chronicles

Jose Aguilar and Kristell Aguilar

CEMISID, Dpto. De Computación, Facultad de Ingeniería
Universidad de los Andes
Mérida, Venezuela

José Gutiérrez de Mesa

Dpto. Ciencias de la Computación, Edificio Politécnico-Campus Universitario
Universidad de Alcalá
Alcalá de Henares (Madrid), Spain

Abstract

The Chronicles are patterns characterized by observable events, with temporal relationships between them. In this work, we propose a model for the building of chronicles, through a learning strategy that allows defining its structure. Our approach discovers the events that compose a chronicle and the temporal relationships between them. These events are defined by changes in the descriptors/features of the modeled phenomena, according to the sequence in which they appear. We test our approach for modeling a hierarchical pattern of vehicle driving styles, which consists of three levels, one to describe the emotional states, another to describe the driver states and, finally, the last one to describe the driving styles. Finally, our approach is compared with other techniques, in classical benchmark classification problems.

Keywords: Learning Algorithms, Chronicles, Recognition Problems

1 Introduction

The chronicles are a paradigm that allows the definition of temporal patterns that represent the behavior of the observed system, from the events that occur in a certain given moment. One of the most important problems with the use of the chronicles is to be able to define them automatically, from the behavior of the system under study. In this work, the main problem studied is the learning of a set of chronicles, in order to represent different situations of the observed system.

The automatic definition of chronicles is a learning problem of great interest. In the literature have been defined two types of learning processes for the paradigm of chronicles [2]: to learn the structure of a chronicle, or to parameterize a generic chronicle. Although these two types of learning have been defined, in the literature, there are practically no proposals developed for them. Normally, experts are used for the design of the chronicles to be used in a study context.

A preliminary work is proposed in [9], where is analyzed the problem of knowledge acquisition in a complex dynamic system. They consider data stream and propose algorithms for analyzing the data logs of alarms: the first stage discovers frequently occurring temporally-constrained alarm sets (called chronicles), and the second stage filters them according to their interdependency. Also, in [6] is proposed a model-based framework about the cardiac electrical activity, where an online analysis of the ECG signals is performed by a chronicle recognition system. They collect time-stamped events since the ECG signals, and use an inductive learning program to build chronicles. In [11], they propose a method based on Petri nets, in order to generate a set of chronicles necessary to the diagnosis, using the fault model of the system to be diagnosed. In [7] is proposed the discovering of temporal patterns hidden in a sequence of events. Particularly, they discover the chronicles hidden in the interaction traces of a human activity, in order to characterize this activity. Finally, in [16], they consider the problem of chronicle learning, based on the analysis of event sequences. They define one single event sequence since multiple event sequences, in order to build a chronicle.

In this work, we propose a new approach for the structural learning of chronicles based on labeled data stream. Our approach is different with respect to previous works because we directly analyze labeled data, in order to determine the events (they are inferred) for a given situation, to build its chronicle. In this way, we propose the definition of a learning mechanism based on labeled data stream, in order to learn the structure of chronicles. It consists of three steps: the creation of the chronicles; the evaluation of the created chronicles; and the confirmation of the chronicles. We propose a set of experiments to test our approach. The first one is the recognition problem of emotions present in a vehicle driver. We use the hierarchical pattern of driving style of vehicle drivers, composed of three levels, defined in [3]. The second one is based on a database of classification problems.

This paper is organized as follows: section 2 presents the chronicles; section 3 presents our structural learning approach of chronicles. Section 4 details the first case study, the metrics used to evaluate the performance, the experiments, the analysis of the results, and the comparison with other works. Section 5 presents the classification experiments, in order to compare with other works. Finally, conclusions and future works are presented.

2 Chronicles

The chronicles are defined by a set of events with temporal constraints between them, representing an interpretation of what is occurring in the dynamics of the system under study at a given time [2]. A chronicle C is defined as a pair (E, T), where E is the set of events and T is the set of temporal constraints between them. Normally, the relationships between the events are temporal, but can be logical [2].

In general, a model that represents a chronicle consists of [4]: i) A set of *times*, which are the temporal components of the chronicle; ii) A set of observed *events*, since the outside world in a given time interval; iii) A set of *temporal restrictions* on which events must occur; iv) A set of *affirmation patterns* (assertions), which describe the context (events, durations, etc.) in which a chronicle occurs at a given time; v) *Actions* to execute when a chronicle is recognized.

The representation of the chronicles is based on propositional terms related to time, and other propositions, making use of predicates. The predicates used in the chronicles are key concepts of this paradigm are [4]: i) hold(P : v, (t1, t2)): establishes that an attribute P maintains the value v in the interval [t1, t2], ii) event(P : (v1 , v2), t): establishes that the attribute P changes its value from v1 to v2 at time t; iii) event(I, t): establishes that the event I occurs in the instant t; iv) noevent(I, (t1, t2)): establishes that there does not happen the event I in the interval [t1, t2]; and v) occurs((n1, n2), I, (t1, t2)): establishes that the event I in the interval of time [t1, t2]], occurs a number of times according to the interval [n1, n2].

It is possible to specify the execution of any action, or external effects to generate, produce new events, when the recognition of an instance of the chronicle happens. To recognize a chronicle is to match the input events (flow) with the events specified in the chronicle (E), respecting the temporal restrictions between them, specified in the chronicle (T). To do this, it must be analyzed the sequences of input events together with their occurrence times, in order to instantiate all the chronicles that match their events with those of the sequence. The recognition of chronicles normally is carried out by a tool to recognize chronicles, called *Chronicle Recognition System (CRS)*, which was developed by Dousson [4], [8]. Two extensions have developed, called CarDeCRS [12] and Matrac [12], which allow the semi-distributed recognition of chronicles. Recently, we have proposed in [4] an extension based on the language CQL, which uses mathematical operators to define the restrictions between the timeless variables and distributed chronicles.

3 Structural Learning Model in the Chronicles

In general, in a real system, we could define generic chronicles for the problem to analyze, and then parameterize these chronicles for a specific context. This approach would require a robust chronicle database, generated from an exhaustive process of extracting of knowledge with experts. The other mechanism is through a learning strategy that uses the available data, to discover/extract the structure of the chronicles. This is the approach that we propose in this work. In particular, we suppose a set of labeled data stream, which is used to describe each class/situation in the system with a chronicle. Our approach to learning chronicle patterns consists of three steps:

A. Creation of Chronicles

In this phase is determined the structure of the chronicles using the algorithm of the Table 1, and the definitions 1, 2 and 3. This algorithm receives a flow of labeled data that describes the different values of the descriptors/features over time, for each class defined by each labeled data. Then, it determines the sequence of events in the system for each class (label), according to the definition 1.

Definition 1: Sequence of Events. It is the flow of events in the system for a given class, based on the behavior of the different descriptors//features that describe it. Every time a value changes in a descriptor, it is an event, and defines a change in the current situation in the system. An event is defined as:

$$E_l^k = \{e_{r_l}, \dots, e_{m_l}\} \qquad (1)$$

Where, E_l^k is the event l of the class (label) k composed of the descriptors D_r … D_m, which have changed for the occurrence of this event. The stamped time in the flow of data determines the time of occurrence of the event. The descriptors in a system are defined as:

$$D_i = \{e_{i_j}, e_{i_{j+1}}, \dots, e_{i_n}\} \qquad (2)$$

Where, D_i is the descriptor i, and, e_{in} is the new value of the descriptor i in the j instant.

In this way can be defined simple events (where only a descriptor changes) and complex events (where several descriptors change). The next step of the algorithm 1 is the definition of the temporal relationships between the events of the same class. For that, the algorithm customizes the times between events, starting from the instant in which they appear in the flow of data. For the determination of the temporal relationships, the algorithm 1 uses the next definition.

Definition 2: Relationship between events. It is the time between the occurrences of two events, which is defined according to eq. 3.

$$T_{dif^k_{y,x}} = t_{E^k_y} - t_{E^k_x} \qquad (3)$$

Where, $T_{dif^k_{y,x}}$ is the temporal difference between the occurrence of the events E^k_y and E^k_x in the class k, and $t_{E^k_y}, t_{E^k_x}$ are the time of the occurrences of the events E^k_y and E^k_x.

Now, our algorithm must generalize the time of occurrences between two events. For that, it uses the different instances of occurrence in the flow of data of the same events in the same class (label), and determines the time interval between these occurrences using the next definition.

Definition 3: Time interval. It is the lapse of time between that an event can occur with respect to another event, in a given pattern.

$$In^k_{y,x} = [Tmin^k_{y,x}, Tmax^k_{y,x}] \qquad (4)$$

Where, $Tmin^k_{y,x}$is the minimum time between the occurrence of the events E^k_y and E^k_x in the class k, and $Tmax^k_{y,x}$is the maximum time between the occurrence of the events E^k_y and E^k_x in the class k.

The output of the model is a chronicle for each label (class) with the sequence of events and their temporal relationships (see Table 1 for more details of the algorithm). In general, for a given class, our algorithm can create different chronicles.

Table 1. Algorithm for creating chronicles

```
Algorithm (for label k)
1.  Repeat for all the registers for the class k
    1.1 Request a register from the database with its stamped time.
    1.2 Detect events based on definition 1
    1.3 Build sequences of events based on Eq. 1 (Build $E_l^k$)
    1.4 Detect the relationships between events in $E_l^k$ using definition 2, and include them on
        the relationships list,
2.  Repeat for all the similar sequences of events of the class k
    2.1 Determine the time interval for the events in the relationship list, using definition 3
```

B. Evaluation of the created Chronicles

The second step determines if the created chronicles can be added to the database of unconfirmed chronicles. For this, we define two operators, as follows [3, 15]:

Definition 4: Inclusion (⊂) between chronicles. A chronicle A is included in a chronicle B if the sequence of events of A, with their respective temporal relationship, forms part of the sequence of events of B.

A generic chronicle is a chronicle with the same sequence of events, but with a temporal relationship between the events that includes a set of other chronicles [3]. Table 2 shows the procedure for the evaluation of a chronicle E:

Table 2. Algorithm for the evaluation of a chronicle

```
Algorithm (Chronicle E)
1 If $E \subset E'$ and $E' \in$ database of chronicles then
    1.1 Update trust(E')=trust(E')+1
2 If $E' \subset E$ and $E' \in$ database of chronicles then
    2.1 Include E in the chronicle database to confirm, mark $E'$ to possibly eliminate if E is confirmed, and trust(E)=trust(E')+1
3 If $E$ is different from the chronicles in the database of chronicles
    3.1 If $E \subset E'$ and $E' \in$ database of unconfirmed chronicles then
        3.1.1 Update trust(E')=trust(E')+1
    3.2 If $E' \subset E$ and $E' \in$ database of unconfirmed chronicles then
        3.2.1 Include E in the database of chronicles to confirm and eliminate $E'$, making trust(E)=trust(E')+1
    3..3 If the chronicle $E$ is different from the chronicles of the database of unconfirmed chronicles
        3.3.1 Include E in the database of chronicles to confirm
```

C. Confirmation of Chronicles

The last step consists of moving the chronicles of the database of chronicles for confirming to the database of confirmed chronicles. The database of chronicles for confirming is composed of chronicles that must be confirmed. This confirmation is carried out in two ways: i) It is defined a threshold that establishes the minimal number of times that a chronicle must happen, and when this number is exceeded, it is automatically confirmed; ii) An expert can determine that it is useful. A confirmed chronicle is moved to the chronicle database, and all the chronicles marked by it in this database of chronicles are eliminated (it includes them).

4 Experiments: Recognition Problem of Emotions

A. Case Study

Our case study is a hierarchical pattern of the style of driving that has been proposed previously [3], which is composed of three levels, one to recognize the emotional states, another to recognize the driver states, and the last one the driving styles. The output of the third level (recognized emotions) feeds the second and first levels; and the status of the recognized driver is used by the first level. Specifically, each level is described as follows (For more details, see [3]):

- **Driving Style level:** it models the driver's way of driving. In the literature, the classical driving styles are [4]: aggressive, ecological, urban and normal. The descriptors of the driving style are: Type of route, State of the driver, Emotion of the driver, Environmental conditions, State of the route, and Characteristics of the traffic.
- **Driver's State level:** it describes the possible state of the driver of a vehicle. According to the literature, the driver's condition can be [13]: awake, stressed, lethargic, pleasant, tired, concentrated, calm, impatient, bored, and asleep, among others. The descriptors used to describe the driver's status are [3]: Class of vehicle, Control actions over the vehicle, Emotion of the driver, Vehicle conditions, Driver characteristics, Driver experience and Driving time.
- **Driver's Emotion level:** it defines the patterns of the driver's emotions. In particular, five of the basic emotions are considered [10]: happiness, sadness, fear, anger, and surprise. The descriptors used to describe them are [3]: Driver’s attention, Control actions over the vehicle, Physiological behavior of the driver, Vehicle conditions, Driver’s voice expressions, and Driver’s facial expressions.

The same situation (an emotion, a driving style, etc.) could be described by different chronicles, to express the diversity of contexts where the same situation can occur (for example, an aggressive behavior). In our case, the variation of the values of a descriptor in a given moment produces an event.

B. Metrics and Scenarios

In order to evaluate the adaptability capabilities of our learning algorithm, we have used several classical metrics defined in the literature [1]: *Precision*, which is defined as the fraction of correct learned instances among the learned instances; *Recall*, which is the fraction of the learned/recognized instances of the pattern, over the total instances that must be recognized. *F-measure* is a measure that combines precision and recall, and *Learning error,* which is the difference between the outputs after the learning process and the outputs (class) that should give. Additionally, in this experiment, we like to verify the next scenarios: a) If our learning algorithm can correctly structure chronicles for the recognition of the emotions, states, and/or styles of a driving; b) If our learning algorithm can define different chronicles to describe different patterns of a same emotion, state, and/or style of a driving.

C. Results of our learning algorithm

Table 3 shows an example of the partial results obtained for the first case, related to learn different patterns for different emotions, states and/or driving styles. Table 4 shows an example of the partial results of our learning algorithm to learn different patterns in the same case, in this example, for the anger emotion. The output of our chronicle learning algorithm consists of several parts (see Tables 3 and 4): the *list of the values of the descriptors* to be evaluated to find the events, the *list of the existing relationships* between the identified events, and the *list of the time intervals* between the events. The list of descriptors contains the registers since the artificial database needed to discover the chronicle (they are grouped by the label of the pattern to be discovered), i.e., each register is a vector where each element (field) represents the value in the register of each descriptor. Also, each register has a time of occurrence in the database. The list of existing relationships is found by our algorithm using the algorithm of the Table 1. It is composed of the ID of the element in the list, the values that have changed between one register and the next, and the ID of the descriptors that have changed (they define the events). Also, it contains the difference of time between the registers. Finally, the list of intervals of time shows the temporal relationship between adjacent events. In particular, Table 3 shows one of the results obtained to learn the ecological style.

Table 3. Case of learning of different patterns

Test Case Title	Learning different patterns
Result for Ecological Style	List of received descriptors ---------------------------------- Element: 1 Values: [1181603, 1, 2, 10034008603, 20453, 3] Time: 195101 Element: 2 Values: [1171803, 1, 1, 10034008603, 20451, 3] Time: 195106 Element: 3 Values: [1167903, 1, 1, 10034008603, 20451, 3] Time: 195111 List of existing relationships ---------------------------------- ID: 1 Changed values: [1171803, 1, 20451] Column of values: [4, 6, 8] Change time: 5 ID: 2 Changed values: [1167903] Column of values: [4] Change time: 5 Time intervals for relationships ----------------------------------- Relationship: [1181603, 2, 20453]<->[1171803, 1, 20451] ColumnR: [4, 6, 8]<->[4, 6, 8] Tchange: 5 Relationship: [1171803]<->[1167903] ColumnR: [4]<->[4] Tchange: 5

Table 4 presents the output of our algorithm for the second case. It shows the learning results for the anger emotion in the same context.

Table 4. Results of learning of different patterns for the same emotion

Test Case Title	Learning different patterns for the same emotion, state or driving style
First instance	
Result for Emotion Anger	List of received descriptors ----------------------------------- Element: 1 Values: [13, 97080109760, 2, 208902, 8, 2] Time: 124651 Element: 2 Values: [11, 97080110770, 2, 208201, 10, 2] Time: 124652 Element: 3 Values: [21, 97080115780, 2, 207952, 10, 2] Time: 124653 List of existing relationships ----------------------------------- ID: 1 Changed values: [11, 97080110770, 208201, 10] Column of values [4, 5, 7, 8] Change time: 1 ID: 2 Changed values: [21, 97080115780, 207952] Column of values: [4, 5, 7] Change time: 1 Time intervals for relationships ---------------------------------- Relationship: [13, 97080109760, 208902, 8]<->[11, 97080110770, 208201, 10] ColumnR: [4, 5, 7, 8]<->[4, 5, 7, 8] Tchange: 1 Relationship: [11, 97080110770, 208201]<->[21, 97080115780, 207952] ColumnR: [4, 5, 7]<->[4, 5, 7] Tchange: 1
Second instance	
Result for Emotion Anger	List of received descriptors ------------------------------------ Element: 1 Values: [22, 96370112820, 1, 201001, 3, 4] Time: 124638 Element: 2 Values: [11, 96370112830, 1, 201101, 3, 4] Time: 124639 List of existing relationships ----------------------------------- ID: 1 Changed values: [11, 96370112830, 201101] Column of values: [4, 5, 7] Change time: 1 Time intervals for relationships ----------------------------------- Relationship: [22, 96370112820, 201001]<->[11, 96370112830, 201101] ColumnR: [4, 5, 7]<->[4, 5, 7] Tchange: 1

D. Building of the chronicles

The results obtained represent the chronicles discovered. They define the structure of the chronicles, as a tree of relationships between the events. The chronicle of Figure 1 is the pattern of the first instance of the anger emotion (see Table 4), discovered by our algorithm.

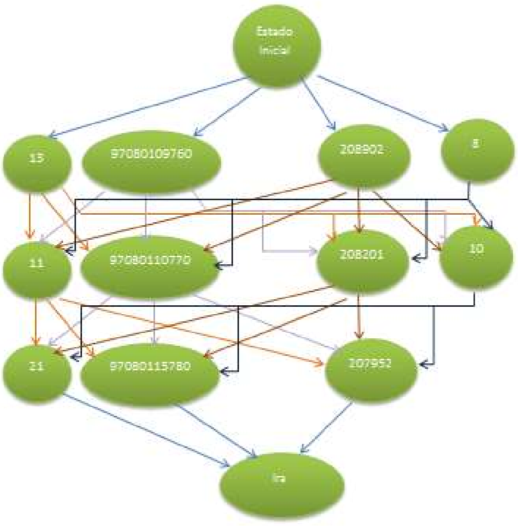

Figure 1. Chronicle created for the anger emotion (first instance of the Table 6) by our algorithm

The tree of relationships is a directed graph, where the root is the previous state of the conductor under study. The tree levels represent each detected event, which can be simple or complex, defined by a set of descriptors whose values changed at that level. That means that a descriptor is in a level if its value changed, allowing the generation of complex events (when change several descriptors) or simple events (when a single descriptor changes). The leaf node represents the learned emotion, state or style, in this case, the anger emotion ("ira" in Spanish). In Figure 1 is shown the anger pattern composed of three levels, where each node represents the values of the descriptors that change, and the arcs connect all the nodes of a level with the nodes of the next level (to indicate the temporal relationship between them). The definition of the time intervals of the temporal relationships is carried out according to the algorithm of the Table 1. Each level of the tree of the Figure 1 is described in the next way:

- *Level 1:* it represents an initial situation (*Ei)*, it symbolizes when the attention of the driver takes the value of 13, the physiological behavior of the driver takes the value 97080109760, the control action on the vehicle takes the value 208902, and the driver's facial expression takes the value 8.
- *Level 2:* it represents a complex event (*Ei2*), because it contains the change of several descriptors. It symbolizes when the driver's attention takes the value of 11, the physiological behavior of the driver takes the value 97080110770, the control action on the vehicle takes the value 208201, and the driver's facial expression takes the value 10. Its occurrence is 4000ms after the event *Ei1*.
- *Level 3*: it represents a complex event (*Ei3*), because it contains the change of several descriptors. It symbolizes when the attention of the driver takes the value of 21, the physiological behavior of the driver takes the value 97080115780, and the action of control on the vehicle takes the value 207952. Its occurrence is 7000ms after event *Ei2.*

In the same way, our learning algorithm constructs more patterns (chronicles) for this emotion, or other patterns, to be learned according to the hierarchy. Now, with each learned chronicle in this phase, our algorithm executes its second and third phases, in order to update the chronicles database.

E. Comparison with other works

In this section, our learning algorithm of chronicles is compared with two other schemes proposed in [3]. The first is called Ar2P (Recursive Pattern Recognition Algorithm), it is a pattern recognition algorithm inspired by the theory of pattern recognition of the mind [14]. The second is based on a fuzzy system of rules of type *IF <Condition> THEN <Action>,* which defines the control actions according to the conditions of the problem that are fulfilled in a given time. In particular, a Multilayer Diffuse Classifier System [5] was used, that consist of a series of fuzzy systems of rules connected hierarchically. In this case, we compare the learning capability of these three techniques. Table 5 shows the results of these techniques for a set of 80 examples from the artificial database.

Table 5 shows the average of the results. The chronicle paradigm has the best behavior. It has the best levels of learning, because it has the best precision and recall (like the algorithm Ar2P), but with a minor error. The fuzzy logic performs a stochastic learning process that is not efficient, while the learning algorithm of Ar2P is quite good, although it converges in superior values to the chronicle. Our algorithm learns exactly the patterns that are derived from the stream of data. In the case of fuzzy logic, in general, the results are less efficient because their learning processes are not efficient and are slow.

Table 5. Comparison with other paradigms

Paradigm	**Learning**			
	Precision	**Recall**	**f-measure**	**Error**
Fuzzy System	0,79	0.82	0.79	0.69
Chronicles	1	1	1	0.03
A2rP	1	0.91	0.95	0.13

5 Experiments: Classification Problems

Now, we test our approach in a classification problem of the UCI Machine Learning Repository. Table 6 describes the characteristics of the dataset used. We select the 70% of the instances for the learning process, and the rest to test the quality of the approach to recognize the classes. The evaluation metric of the recognition process is the precision.

Table 6 Dataset of the classification problem

Dataset	Size	Descriptors	Characteristics
Human Activity Recognition Using Smartphones	10299	561	six activities/classes (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II)

The results of the test phase are shown in Table 7. The precision is similar and there is a small difference between our proposal and the algorithm with the best precision. The main point is that our algorithm can obtain the patterns for the classes. These patterns can be used to recognize the classes with a similar quality of the classical classification algorithms, and additionally, each pattern has a structure that describes the classes according to the events that our algorithm discovers. These patterns are not generated by the classical classification algorithms, which can be interesting in some problems where the structure of the patterns is important to know.

Table 7. Precision of the different algorithms

Algorithm	Human Activity
RBF	0.889
Logistic	0.944
MultilayerPerceptron	0.958
Our algorithm	0.9353

6 Conclusions

The process of adaptation of the chronicles allows modeling the same situation with different chronicles, expressing the diversity of contexts in which the same situation can occur. In general, our algorithm automates the process of generating of the chronicles since a labeled data stream. Our algorithm allows discovering chronicles, and makes possible to learn the temporal relations existing between the set of events of the chronicles. The developed algorithm is flexible, since it can adjust to new descriptors and new situations in real-time.

Initially, we have studied a complex problem, the learning of a hierarchical pattern of vehicle driving styles, and we have carried out a comparative analysis with two techniques, fuzzy logic, and Ar2P algorithm. Our learning approach presents better results than the model based on fuzzy logic in all the criteria, and results very similar that Ar2P. Additionally, we have tested our algorithm in a classical classification benchmark problem, and we have obtained very good precision results. In addition, it gives a structure of the patterns for each class that can be used for analysis tasks, in order, e.g., to define causes (diagnosis), or descriptors with a main role. The classical classification algorithms only solve the classification problem. In general, our paradigm obtains very good results in all cases, and additionally, it gives a lot of information about the learned situations, because it defines patterns based on temporal logic that can be used to infer things.

Future works are going to explore the utilization of this algorithm in smart environments, in particular, we are interested in smart classrooms, where it is required the recognition of the emotions of the students [15, 17]. Also, we need to extend this algorithm for the case of flow of unlabeled data (for the case of clustering) and hybrid learning problems, and integrate it with an approach to parameterize generic chronicles. Finally, we need to explore how integrate sets of confirmed chronicle databases of different problems, in a federation of chronicle databases.

References

[1] J. Aguilar, Learning Algorithm and Retrieval Process for the Multiple Classes Random Neural Network Model, *Neural Processing Letters*, **13** (2001), no. 1, 81-91. https://doi.org/10.1023/a:1009611918681

[2] J. Aguilar, *Temporal Logic from the Chronicles Paradigm: Learning and Reasoning Problems, and Its Applications in Distributed Systems*, LAP Lambert Academic Publishing, 2011.

[3] J. Aguilar, K. Aguilar, D. Chavez, J. Cordero, E. Puerto, Different Intelligent Approaches for Modeling the Style of Car Driving, *14th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Vol. 2, (2017), 284-291. https://doi.org/10.5220/0006411902840291

[4] J Aguilar, E. Exposito, A. Subis, J. Vizcarrondo, ARMISCOM: Autonomic Reflective Middleware for management Service composition, *4th Global Information Infrastructure and Networking Symposium*, (2012). https://doi.org/10.1109/giis.2012.6466760

[5] E. Camargo, J. Aguilar, Hybrid intelligent supervision model of oil wells, *IEEE International Conference on Fuzzy Systems*, (2014), 934–939. https://doi.org/10.1109/fuzz-ieee.2014.6891725

[6] G. Carrault, M. Cordier, R. Quiniou, M. Garreau, J. Bellanger, A. Bardou A Model-Based Approach for Learning to Identify Cardiac Arrhythmias, Chapter in *Artificial Intelligence in Medicine*, Springer, 1999, 165-174 https://doi.org/10.1007/3-540-48720-4_18

[7] D. Cram, B. Mathern, A. Mille, A complete chronicle discovery approach: application to activity analysis, *Expert Systems*, **29** (2012), no. 4, 321-346. https://doi.org/10.1111/j.1468-0394.2011.00591.x

[8] C. Dousson, Extending and Unifying Chronicle Representation with Event Counters, *15th Eureopean Conference on Artificial Intelligence*, (2002).

[9] C. Dousson, T. Duong, Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems, *16th International Joint Conference on Artifical intelligence*, Vol. 1, (1999), 620-626.

[10] P. Ekman, W. Friesen, Constants across cultures in the face and emotion, *Journal of Personality and Social Psychology*, **17** (1971), no. 2, 124-129. https://doi.org/10.1037/h0030377

[11] B. Guerraz, C. Dousson, Chronicles construction starting from the fault model of the system to diagnose, *International Workshop on Principles of Diagnosis*, (2004), 51–56.

[12] X. Le Guillou L. Rozé, CarDeCRS and Matrac: two chronicle-based distributed diagnosis platforms, Technical Report, INRIA. https://goo.gl/RB1Ajp

[13] C. Lisetti, F. Nasoz, Affective intelligent car interfaces with emotion recognition, *11th International Conference on Human Computer Interaction*, (2005).

[14] E. Puerto, J. Aguilar, Learning Algorithm for the Recursive Pattern Recognition Model, *Applied Artificial Intelligence*, **30** (2016), 662–678. https://doi.org/10.1080/08839514.2016.1213584

[15] M. Sánchez J. Aguilar, J. Cordero P. Valdiviezo, A Smart Learning Environment based on Cloud Learning, *International Journal of Advanced Information Science and Technology*, **39** (2015), 39-52.

[16] A. Subias L. Travé-Massuyès E. Le Corronc, Learning chronicles signing multiple scenario instances, *IFAC Proceedings Volumes*, **47** (2014), 10397-10402. https://doi.org/10.3182/20140824-6-za-1003.02579

[17] P. Valdiviezo, J. Cordero J. Aguilar, M. Sánchez, Conceptual Design of a Smart Classroom Based on Multiagent Systems, *Int. Conf. Artificial Intelligence*, (2015), 471-477.

Received: February 25, 2018; Published: April 10, 2018