

Intelligent Agents and Complex Event Processing to enhance Patient Monitoring

Lorenzo De Lauretis^{1,†}, Fabio Persia^{1,*} and Stefania Costantini^{1,*}

¹*Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Via Vetoio, L'Aquila, 67100, Italy*

Abstract

This paper describes a smart ecosystem able to continuously monitor patients' health condition, at home or at work or during recreational activities, by exploiting a creative blend of Medical Wearables, Intelligent Agents (IA) and Complex Event Processing (CEP). With the help of a smart application, that links together the Wearables and the power of IA and CEP, patients will be continuously and actively supervised during their daily activities. This can even save their lives, in case sudden or gradual issues should occur. Thanks to our system, patients with non-severe though potentially unstable chronic diseases will no longer overburden first-aid services. This is also useful for containing the spread of COVID-19. Specifically, in this paper we focus on automated vitals monitoring, electrocardiogram (ECG) analysis, and psoriasis detection. Experimental results carried out on real patients show how promising our approach is.

Keywords

Intelligent Agents, Complex Event Processing, Event Query Language, Interval Relationships, Wearables, Image Processing

1. Introduction

The Coronavirus pandemic has highlighted telehealth as a crucial component of modern and sustainable treatment. In fact, telemedicine's primary purpose is to virtually erase the distance between patient and physician, as well as to reduce time and expenses involved in healthcare access. Consequently, in this context here we propose an approach which exploits in a synergistic way wearable devices (such as portable ECG devices and pulse oximeters), Intelligent Agents (IA), and Complex Event Processing (CEP) techniques, to create an integrated framework able to follow the patient wherever (s)he is.

In the literature, even if some works may appear similar to ours [1], they implement stand-alone solutions with a single "brain", monitoring a limited number of situations; in our work, instead, we created a smart ecosystem, with more "brain-sectors" linked together, each one

Datalog 2.0 2022: 4th International Workshop on the Resurgence of Datalog in Academia and Industry, September 05, 2022, Genova - Nervi, Italy

*Corresponding author.

†These authors contributed equally.

✉ lorenzo.delawaretis@graduate.univaq.it (L. D. Lauretis); fabio.persia@univaq.it (F. Persia); stefania.costantini@univaq.it (S. Costantini)

🌐 <https://www.disim.univaq.it/FabioPersia> (F. Persia)

🆔 0000-0002-1798-0215 (F. Persia)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

📄 CEUR Workshop Proceedings (CEUR-WS.org)

specifically needed for some kind of calculations, that, together, have a very strong computational power.

The remainder of the paper is organized as follows. Section 2 describes the system architecture, section 3 shows an overview of the experimental results, and section 4 concludes the paper and outlines future work.

2. System Architecture

Our system consists of six main components: User and Doctor Interfaces - Android App, Database and Web interface, Hardware, Intelligent Agents, Complex Event Processing and Image Processing.

Figure 1 depicts the entire workflow of our system. Specifically, to interact with the system, first of all the patient (possibly supported by a caregiver) downloads the Android application on his/her smartphone. After downloading the app, (s)he can begin using it, inserting his/her personal information, such as name and age. Then, (s)he can perform the *manual data acquisition*, by entering the information listed in the so called activity within the green box in Figure 1.

After inserting all the manual data, and after wearing suitable devices if needed, the patient can proceed to the actual usage of the system.

In all situations, if, after analyzing the data in connection to the sensations and wellbeing impressions possibly provided by the patient, the patient turns out to be healthy, then our system does nothing, except for providing reassurance and suitable advice (e.g., “stay seated for some minutes, breath deeply and try to relax”), and reminding the patient of the medicines to take. If minor issues are detected, then the system alerts the user using Android notifications. If severe issues are detected, then the system alerts the doctor, and, if the situation is critical, it alerts the first aid services.

Oxygen saturation is obtained via the Arduino device. The device is connected via Bluetooth, so the data are directly sent to the server through the application.

The manual data and the oxygen saturation value are then sent to the DALI Intelligent Agents, which proceed to searching for and identifying possible health issues. The agent coding has been performed following the advice of medical doctors.

After obtaining oxygen saturation, the patient can start the ECG acquisition process, by connecting the related ECG device. After connecting, wearing the device, and acquiring the data, those are automatically sent to the server for processing.

The ECG data are then elaborated, and short-term events, if present, are detected; the data are subsequently sent to the ISEQL engine, which is in charge of constantly checking for the presence of health issues that cannot be seen by instantaneous analysis, but can be instead detected via long-term observation, elaboration and aggregation of events (this by performing “Complex Event Processing”).

Additionally, the patient can also send pictures of his/her face and/or hand to be checked by a doctor, in case (s)he wants her/him to better evaluate the situation. In the case of the face picture, the image is directly sent to the doctor, who, after evaluating the patient’s appearance, can decide, if the detected situation does not look good (e.g., excessive paleness with deep dark circles), to alert either the user, providing suitable advice or making an appointment for a

face-to-face medical examination, and/or the first aid services. In the case of the hand picture, it is automatically processed by our software, to find signs of hand disease. In particular, as a first case study, we have considered *psoriasis* as it is a rather frequent pathology.

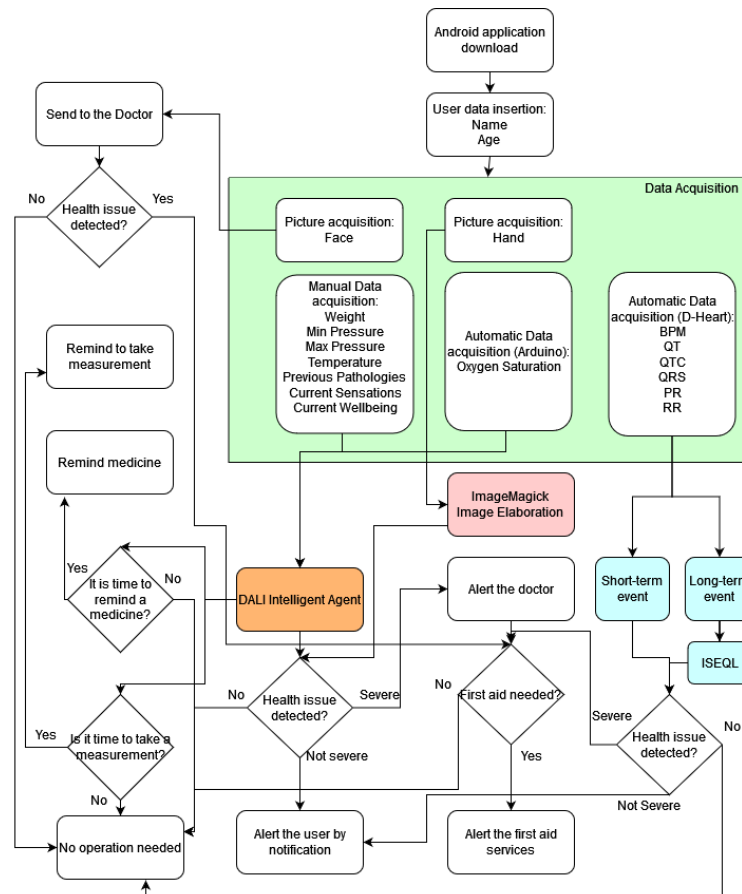


Figure 1: System workflow

2.1. Intelligent Agents

In our work, we use DALI [2] in order to implement a system composed by intelligent agents. DALI is an agent-oriented logical language derived from Prolog, and, similarly to it, is based on the logic programming paradigm. It has been fully implemented, on the basis of a fully logical semantics. Exploiting the DALI language, we defined two intelligent agents: the *Patient agent* and the *Doctor agent*. The Patient agent analyses the patient’s vital parameters (heart rate, saturation, minimum and maximum pressure, temperature, weight) taken by wearable devices or by the patient, and returns immediate feedback on each of them.

It also acts as an e-health companion, reminding the user to take pills at the correct time, sending a reminder for temperature, blood pressure and weight measurements. The Doctor agent receives messages from the Patient agent, analyzes the communicated problem, e.g., fever

or tachycardia, and responds by suggesting a medication to take or, in case of serious danger, it suggests going to the Emergency Room. Events currently detected by our DALI intelligent agents are *angina pectoris*, *tachicardia*, *bradycardia*, *hypoxia*, *hypertension* and *fever* (Figure 2).

```

angina_ev(X) :- after_evp_time(sensation(_,0,0,0,1), angina(X).
angina_evi(X) :- assert(danger(X)), retract(angina(X)).

whats_up :- after_evp_time(angina_ev(_,0,0,10,0).
whats_upI :- Message = 'how', mas_send(Message).

tachycardia_ev :- after_evp_time(bpm(_,0,0,0,1), lists(heartbeat,LH),
last_values_mag(LH,120,Res), Res == 'si'.
tachycardia_evi :- assert(danger(tachycardia)).

bradycardia_ev :- after_evp_time(bpm(_,0,0,0,1), lists(heartbeat,LH),
last_values_min(LH,60,Res), Res == 'si'.
bradycardia_evi :- assert(danger(bradycardia)).

hypoxia_severe_ev :- after_evp_time(saturation(_,0,0,0,1), lists(saturation,LS),
last_values_min(LS,90,Res), Res == 'si'.
hypoxia_severe_evi :- assert(danger(hypoxia_severe)).

presmin_high_ev :- after_evp_time(presminima(_,0,0,0,1), lists(diastolic_pressure,LMin),
last_values_mag(LMin,90,Res),
(Res == si -> print('Pressure minimum high'),nl, true; retract(presmin_high), false).
presmin_high_evi :- assert(presmin_high).

presmin_low_ev :- after_evp_time(presminima(_,0,0,0,1), lists(diastolic_pressure,LMin),
last_values_min(LMin,60,Res),
(Res == si -> print('Pressure minimum low '),nl, true; retract(presmin_low), false).
presmin_low_evi :- assert(presmin_low).

presmax_high_ev :- after_evp_time(presmassima(_,0,0,0,1), lists(systolic_pressure,LMax),
last_values_mag(LMax,140,Res),
(Res == si -> print('Pressure maximum high'),nl, true; retract(presmax_high), false).
presmax_high_evi :- assert(presmax_high).

presmax_low_ev :- after_evp_time(presmassima(_,0,0,0,1), lists(systolic_pressure,LMax),
last_values_min(LMax,90,Res),
(Res == si -> print('Pressure maximum low '),nl, true; retract(presmax_low), false).
presmax_low_evi :- assert(presmax_low).

hypotension_ev :- after_evp_time(presmax_low_ev,0,0,0,2), presmin_low, presmax_low.
hypotension_evi :- assert(danger(hypotension)), retract(presmin_low), retract(presmax_low).

hypertension_ev :- after_evp_time(presmax_high_ev,0,0,0,2), presmax_high, presmin_high.
hypertension_evi :- assert(danger(hypertension)), retract(presmax_high), retract(presmin_high).

i_have_fever(X) :- after_evp_time(temperatura(_,0,0,0,1), fever(X,Y), Y == 'high'.
i_have_feverI(X) :- messageA(doctorAgent, send_message(fever(X),Ag)), retractall(fever(_,_)),nl,
print('Patient has fever at '), print(X).

```

Figure 2: A sample of the code used to detect possible health problems and alarming situations

2.2. Complex Event Processing

The language we exploit in order to carry out Complex Event Processing tasks in this context is named *ISEQL* (it stands for “an Interval-based Surveillance Event Query Language”); some of the authors of this paper defined it in [3] and extended it in [4]. The proposed language extends relational algebra and the well-known Allen’s interval relationships [5], is very flexible and useful to effectively define event models in video surveillance [6] and social network analysis domains, and is also efficiently implemented exploiting a very quick algorithm [7] which is several times faster and scales better than state-of-the-art approaches [8, 9], while being much better suited for real-time event processing. Here we use it to effectively model and detect short-term and long-term events in patient’s ECGs.

2.2.1. Short-term event detection

A *Short-term Event* is an event to be detected within the context of a single ECG measurement. To detect such events, we must analyze the ECG signal in depth using specific criteria stored in

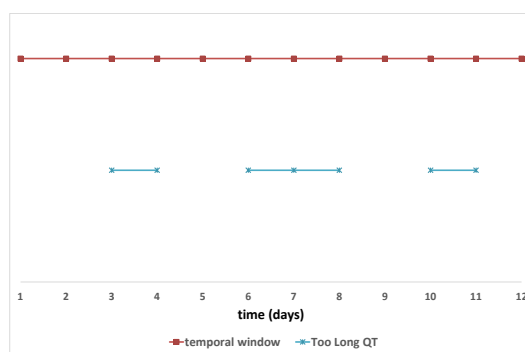


Figure 4: Scenario with many *Too Long QT* ECGs, not always on consecutive days

“visible” to the computer, we exploit one of the most used software for image elaboration, that is named ImageMagick².

3. Experimental Results

Table 1 summarizes the preliminary experimental results carried out. Specifically, each column represents a case study; respectively, a healthy person, a patient suffering from Atrial Fibrillation (AF), and another patient suffering from Wolff-Parkinson-White (WPW) syndrome. Eventually, the last column deals with the case study on psoriasis. All the tests achieved a precision of 1. Three out of four tests obtained a recall of 1, indicating that, for such three tests, the maximum proportion of actual positives was identified correctly. As regards the last test, i.e., the one about psoriasis detection, the system achieved a recall of 0.875, that, even being a high value, it is not the maximum, and thus can be hopefully improved in the near future.

Table 1
Summary of all tests

	Healthy	AF	WPW	Psoriasis
True Positives	30	5	10	7
False Positives	0	0	0	0
False Negatives	0	0	0	1
Precision	1	1	1	1
Recall	1	1	1	0.875

²<https://imagemagick.org>

4. Conclusion

In this paper, we proposed an integrated framework aimed at actively supporting patient monitoring by exploiting an innovative combination of wearables, DALI intelligent agents and Complex Event Processing. The developed framework also includes an effective image processing algorithm for psoriasis detection. Preliminary experiments conducted on real patients confirm the validity of the proposed approach.

Future work will be devoted to extend the system in many ways; in fact, the intelligent agent will cope with many other health issues, the set of short-term and long-term events detectable via ISEQL will be expanded, and the psoriasis detection algorithm will be able to automatically detect other skin diseases.

References

- [1] A. H. Jabber, A. Obied, A multi-agent system in e-health system implementing ebdi model, *Turkish Journal of Computer and Mathematics Education* 12 (2021) 2845–2859.
- [2] S. Costantini, A. Tocchio, About declarative semantics of logic-based agent languages, in: M. Baldoni, U. Endriss, A. Omicini, P. Torroni (Eds.), *Declarative Agent Languages and Technologies III, Third International Workshop, DALT 2005, Selected and Revised Papers*, volume 3904 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 106–123.
- [3] S. Helmer, F. Persia, Iseql, an interval-based surveillance event query language, *International Journal of Multimedia Data Engineering and Management (IJMDEM)* 7 (2016) 1–21.
- [4] F. Persia, S. Helmer, A framework for high-level event detection in a social network context via an extension of ISEQL, in: *12th IEEE International Conference on Semantic Computing, ICSC 2018, Laguna Hills, CA, USA, January 31 - February 2, 2018*, IEEE Computer Society, 2018, pp. 140–147. doi:10.1109/ICSC.2018.00028.
- [5] J. F. Allen, Maintaining knowledge about temporal intervals, *Commun. ACM* 26 (1983) 832–843. URL: <https://doi.org/10.1145/182.358434>. doi:10.1145/182.358434.
- [6] F. Persia, S. Helmer, An event detection framework supported by a smart graphical user interface, in: *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), 2016*, pp. 490–495. doi:10.1109/IRI.2016.72.
- [7] D. Piatov, S. Helmer, A. Dignös, F. Persia, Cache-efficient sweeping-based interval joins for extended allen relation predicates, *The VLDB Journal* (2021) 1–24.
- [8] T. Y. C. Leung, R. R. Muntz, Query processing for temporal databases, in: *ICDE, 1990*.
- [9] Z. Khayyat, W. Lucia, M. Singh, M. Ouzzani, P. Papotti, J.-A. Quiané-Ruiz, N. Tang, P. Kalnis, Fast and scalable inequality joins, *VLDB J.* 26 (2017).
- [10] J. Fayn, L. Conti, S. Fareh, P. Maison-Blanche, P. Nony, P. Rubel, Interactive and dynamic ECG analysis. Is it just an IDEA or a clinically relevant approach?, *J Electrocardiol* 29 Suppl (1996) 21–25.
- [11] B. T. Fengler, W. J. Brady, C. U. Plautz, Atrial fibrillation in the wolff-parkinson-white syndrome: Ecg recognition and treatment in the ed, *The American Journal of Emergency Medicine* 25 (2007) 576–583.