

# The PredictMed-PoMAS Architecture for Intelligent Patient Monitoring within a Complex Healthcare Ecosystem

Stefania Costantini<sup>1,\*</sup>, Carlo Bertoncelli<sup>2</sup>, Domenico Bertoncelli<sup>3</sup>, Pasquale De Meo<sup>4</sup>, Rino Falcone<sup>5</sup>, Fabio Persia<sup>1</sup> and Silvia Rossi<sup>6</sup>

<sup>1</sup>Department of Information Engineering, Computer Science and Mathematics -DISIM- Università di L'Aquila, Italy

<sup>2</sup>Côte-d'Azur University of Nice and Etablissement pour Enfants ou Adolescents Polyhandicapés H. Germain, Fondation Lenval, Nice, France

<sup>3</sup>Department of Computer Science, University of West Florida, USA

<sup>4</sup>Department of Ancient and Modern Civilizations, University of Messina, Italy

<sup>5</sup>Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy

<sup>6</sup>Department of Electrical Engineering and Information Technologies -DIETI- University of Naples "Federico II", Italy

## Abstract

In this paper, we leverage our past work to outline and describe, also using examples, a comprehensive architecture that goes from PMAs, i.e., Patient Monitoring Agents, to a novel notion of Hybrid Society (HS) instantiated in this paper to the healthcare domain, different from previous existing proposals. The HS is meant to encompass PMAs and offices of medical specialists, healthcare companies such as medical centres, and institutions such as hospitals. Our approach is based on approaches to agents and Multi-Agent-Systems (MAS) rooted in computational logic, whose nice formal foundation allows properties of agents and MAS to be proved/enforced through theorem proving, model-checking, run-time verification and other techniques.

## Keywords

Multi-Agent Systems, Prediction Models, Patient Monitoring, Hybrid Society

## 1. Introduction

There are many illnesses and health conditions where timely detection and constant monitoring are of utmost priority for early remediation and prevention. The inability to detect dangerous conditions may cause serious damage and, potentially, irreversible detrimental variations in the organ system that may even lead to a patient's demise. Therefore, developing novel, personalized, affordable clinical monitoring systems is urgently required, and in this effort, many AI (Artificial Intelligence) techniques, by themselves or in combination, can be profitably exploited.

---

HC@AIxIA 2023, 2nd AIxIA Workshop on Artificial Intelligence For Healthcare

\*Corresponding author.

✉ stefania.costantini@univaq.it (S. Costantini); carlolaser777@gmail.com (C. Bertoncelli);

domenico.bertoncelli@gmail.com (D. Bertoncelli); pasquale.demeo@unime.it (P. D. Meo); rino.falcone@ist.cnr.it

(R. Falcone); fabio.persia@univaq.it (F. Persia); silvia.rossi@unina.it (S. Rossi)

🌐 <https://www.disim.univaq.it/StefaniaCostantini> (S. Costantini)

🆔 0000-0002-5686-6124 (S. Costantini)



© 2023 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)

In recent work [1, 2], we showed on specific case studies how multi-agent systems, aided by prediction models based on various forms of Machine Learning, and exploiting the use of wearable devices [3] whose data are aggregated via Complex Event Processing techniques, may indeed constitute the “core” of such a system.

In this paper, abstracting away from special cases, we intend to illustrate the general architecture for patient monitoring that we devised by leveraging upon past work that we call PredictMed-PoMAS. The architecture encompasses Personal Monitoring Agents (PMAs) for patients. Using PMAs, we aim to provide support to: on the one hand, patients, who are constantly monitored by the system, which can provide them (or their caregivers) with support and advice; on the other hand, medical doctors, who are constantly informed about their patient’s conditions and are provided with all the necessary elements to take their clinical decisions. PMAs are constructed in our envisaged system as a combination of the PredictMed and PoMAS components.

The PredictMed component is a prediction model using statistical data mining and machine learning developed and validated by some of the authors of this paper [4, 5, 6, 7].

The Patient Observing MAS (PoMAS) is a novelty within the existing literature, as it includes a Complex Event Processing (CEP) module that, by examining event time series (coming from sensors or the patient/caregivers), provides real-time detection of short- and long-term events related to the patient’s condition, requiring countermeasures to be taken (immediate, in case of short-term events) and doctors to be informed. CEP results allow for more high-level communication with the system’s users and, consequently, a better definition of suitable treatment and intervention. The predictive component is leveraged to provide input to the PoMAS to identify and undertake immediate countermeasures and alert the parties involved to changes in key indicators associated with an increased likelihood of adverse events.

A key feature of PMAs will be a user-friendly interface for patients, doctors, and care providers.

We then enlarge our view and situate PredictMed-PoMAS PMAs as components of a wider healthcare ecosystem that we are defining. This wider environment will encompass all relevant actors of a healthcare system, i.e., for instance, PMAs, offices of medical specialists, healthcare companies such as medical centres, and institutions such as hospitals. Patients’ PMAs can thus locate in such a system all necessary services. At the same time, the overall system can operate based on its own objectives: to reconcile patients’ interests and optimal or near-optimal resource allocation and usage. This ecosystem, in turn, can be seen as an instance of the Hybrid Society (HS) we intend to advocate. The notion of HS that we introduce and propose is different from and more general than what is found in current literature [8, 9]. In the proposed HS, humans and autonomous systems (AS) should be coupled at multiple levels based on shared agreed-upon principles and standards, which must, by definition, enforce tight constraints on agents’ behaviour. Such principles should include values and social norms and ethical and professional conduct codes relative, in the specific case, to the healthcare field. The proposed notion of Hybrid Society (HS) and the envisaged approach to the development of AS and HS will be grounded on three main pillars: 1. Verifiability (trustworthiness) to ensure that all components of the hybrid society, whatever its size, comply with their expected behaviour and that violations will be corrected. 2. Trust. Users are better motivated to embrace a system with a high perceived level of trust in its ethical behaviour. 3. Explainability. Numerous studies have linked trust to the possibility of having a verifiable system that can provide explanations which

are understandable by each specific category of users.

Related existing work concerns the Agentcities project [10]. Agentcities is meant to be a network of FIPA-compliant<sup>1</sup> agent platforms aimed at demonstrating the potential of autonomous agents, which can dynamically discover one another in the network to interact and offer composite services. Special attention is on the architecture, which is, however, centralized and requires, to some extent, human intervention to link new MASs to the network. An instance of the architecture concerns healthcare [11], where agents allow the user to access his/her medical record, find out information about the city's medical centres, and make appointments with doctors. Doctors, in turn, gain access to a patient's medical records during a visit and may also request tests or other visits. In Agentcities, the issues of ethics and trust are mentioned but not elaborated upon; the concept of PMAs is absent; the architecture is not as general as the proposed one; there is no notion of the system's objectives.

We are especially interested, as a base for the HS, in agent-oriented approaches based on Computational Logic [12] because these technologies enable trustworthiness, in the sense that agents should be relied upon to do what is expected of them, while not exhibiting unwanted behaviour. So, agents should not behave in improper/forbidden/unethical ways and should not devise new behaviours that might contradict their specifications or, however, the user's expectations. They should be transparent in the sense of being able to explain their actions and choices when required. In a computational logic setting, trustworthiness can be ensured by various a-priori and run-time verification techniques: cf., e.g., [13, 14] and the references therein. Much work is underway in the field for ethics in agents: consider, for instance, [15, 16] and the references therein.

The paper is structured as follows. In Section 2, we discuss and illustrate the components of our PMAs. In Section 3, we describe the envisaged healthcare-oriented Hybrid Society. All along, we make use of illustrative examples. Finally, in Section 4, we conclude.

## 2. PredictMed-POMAS PMAs

### 2.1. PredictMed

The PredictMed component is a prediction model using statistical data mining and machine learning. The PredictMed logistic regression-based model uses an algorithm implemented in the R programming language. After splitting the patients in training and testing sets, logistic regressions are performed, in principle, on every possible subset (tuple) of independent variables. The tuple that shows the best predictive performance in accuracy, sensitivity, and specificity is chosen as a set of independent variables in another logistic regression to calculate the probability of the single patient developing the specific health condition under observation. However, due to the exponential blowup, this method will fail in clinical settings with many variables. Then, in practice, more sophisticated forms of feature selection that we do not discuss here are employed. The model was developed and validated by some of the authors of this paper on a variety of cases, among which to predict health conditions in children [4, 5, 6] and osteoarthritis in adults

---

<sup>1</sup>FIPA is a widely used standardized ACL (Agent Communication Language), cf. <http://www.fipa.org/specs/fipa00037/SC00037J.html>

[7], and to predict the development of scoliosis, intellectual disabilities, and autistic features. All data were analyzed anonymously from a database, including demographics, functional diagnosis, and neurologic and cognitive assessments. The average accuracy, sensitivity, and specificity score in these cases was 90%. The model can be adapted to predict many other medical conditions. In [2], for instance, we adapted and tested PredictMed to identify factors associated with epilepsy, particularly the immediate danger of having a seizure, in children with certain specific diagnosed neurological disorders. In that application, data from wearable devices (accelerometric or electrodermal data, ECG data, etc.) were processed to extract their most relevant features. Leveraging previous studies, features useful for the PredictMed learning modules were obtained by processing wearable device data. These features become part of a Time Series data stream. By learning on a Training Set of time series data the normal behaviour, the system can find anomalies in the pattern of data when a fixed threshold is exceeded, which in practice constitutes a warning in advance for a seizure to occur. After testing and validation, the system has been used to implement a real-time automated seizure detection system for young patients.

## 2.2. Complex Event Processing

In the proposed architecture, we employ *Complex Event Processing* (CEP) techniques to improve patient monitoring. We, in particular, exploit, for the definition of CEP techniques, the dedicated ISEQL language (Interval-based Surveillance Event Query Language), which some of the authors of this paper defined in [17, 18], based upon the well-known Allen's interval relationships.

From timed streams of events collected by some wearable devices, suitably defined ISEQL expressions can detect both *short-term* events and *long-term* events, complying with the guidelines in the medical literature. For instance, in the analysis of ECGs [19], a *short-term event* is an event to be detected within the context of a single ECG measurement. The events can be modelled in ISEQL, and their severities can be measured on a suitable scale. Working on aggregated data at a different resolution, such as, e.g., examining the results of several ECG measurements taken in specified more expansive temporal windows, allows a detailed overall analysis of the patient's clinical history, thus detecting *long-term* events.

There can be an interaction with PredictMed: the former can learn to detect anomalies in a series of data, thus identifying new short/long-term events that CEP should consider in addition to predefined ones. Then, in perspective, new ISEQL expressions might be, even automatically, added.

As an example, a simple long-term event instance identified by the Complex Event Processing Engine concerning the early detection of possible seizures is shown in Figure 1. The x-axis represents the minutes when the patient is monitored, and the red interval is the specific temporal window we want to take into account for further analyses. Additionally, the other intervals (i.e., Seizure 1, ..., Seizure 4) represent the temporal spans when specific instances of epileptic seizures have been detected by the short-term module. As a consequence, one scenario such as the one in Figure 1 should be immediately identified as a long-term event since there are multiple occurrences of anomalies in a relatively short temporal window. Thus, a medical doctor must be contacted immediately. As we may notice, from a technical point of view, ISEQL allows a designer to easily model and identify these kinds of scenarios.

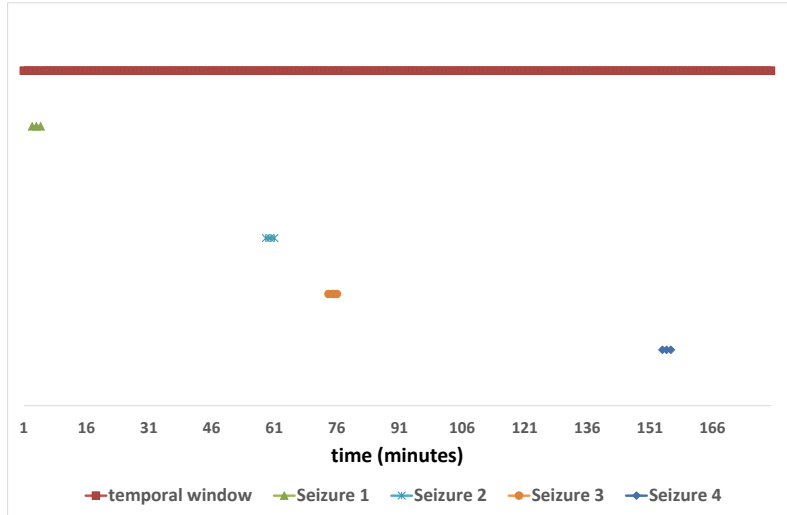


Figure 1: A possible long-term event instance

## 2.3. PoMAS: Patient’s Observation and Monitoring Multi-Agent System

### 2.3.1. Background

Before illustrating the architecture of the PoMAS, we provide some necessary background on Logical Agents and the DALI language that we adopted in our prototypical implementation.

Agents and Multi-Agent Systems (MAS) is a very relevant field of research and application in Artificial Intelligence; as any intelligent system which is meant to be autonomous, whatever its inner nature (even, e.g., a trained Machine Learning device), must be ultimately encapsulated within an agent or a MAS.

A survey on the main approaches to agents is [20]. It can be seen therein that many computational-logic-based agent-oriented languages and frameworks to specify agents and Multi-Agent Systems have been defined over time. Their added value with respect to non-logical approaches is to provide clean semantics, readability, verifiability, transparency, and explainability ‘by design’ (or almost), as logical rules can easily be transposed into natural-language explanations.

DALI [21, 22, 23] is a logical Agent-Oriented Logic Programming language, where the autonomous behaviour of a DALI agent is triggered by several kinds of events: external events, internal, present, and past events.

External events are syntactically denoted by the postfix  $E$ . Reaction to each of such events is defined by a reactive rule, where the special token  $:>$  is used. The agent remembers to have reacted by converting an external event into a *past event* (postfix  $P$ ). An event perceived but not yet reacted to is called as “present event” and is indicated by the postfix  $N$ . It is often useful for

an agent to reason about present events that make the agent aware of what is happening at the current moment in its external environment.

In DALI, actions (denoted by postfix *A*) may have or not have preconditions: in the former case, the actions are defined by action rules; in the latter case, they are just action atoms. The new token `:<` characterizes an action rule that specifies an action's preconditions. Similarly to events, actions that have been performed are recorded as past actions.

"Internal events" are the device which makes a DALI agent proactive. An internal event is syntactically denoted by the postfix *I*, and its description comprises two rules. The first one contains the conditions (knowledge, past events, procedures, etc.) that must be true so that the reaction (in the second rule) may happen. Thus, a DALI agent can react to its own conclusions. Internal events are automatically attempted with a default frequency, customizable using user directives.

The DALI communication architecture [24] implements the DALI/FIPA protocol, which consists of the main FIPA primitives for language specification, syntax and semantics, plus a few new primitives which are peculiar to DALI. Notice that DALI has been made compatible with the Docker technology (cf. [25] for details). So, a DALI agent can be deployed within a container.

The DALI programming environment [23] is freely available and, at the current stage of development, offers a multi-platform folder environment built upon Sicstus Prolog programs, shells scripts, and Python scripts to integrate external applications, a JSON/HTML5/jQuery web interface to integrate into DALI applications, with a Python/Twisted/Flask web server capable of interacting with a DALI MAS at the back-end. We have recently devised a cloud DALI implementation, reported in [26, 27, 25, 28].

### 2.3.2. POMAS

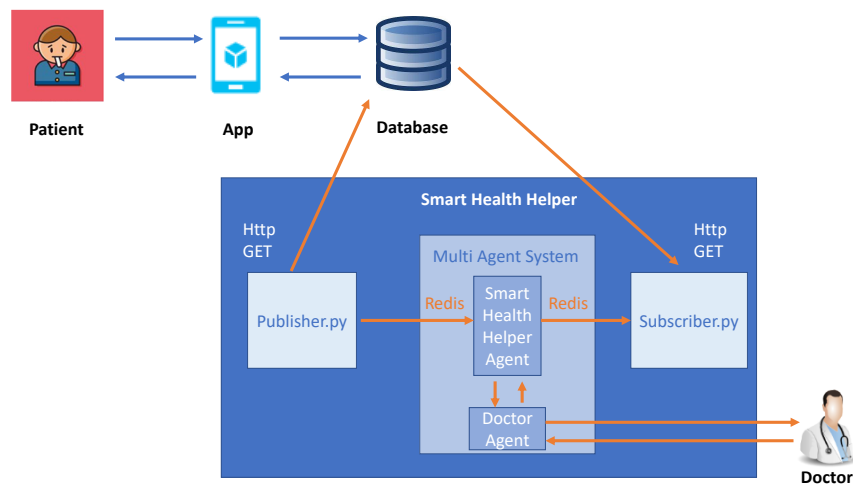
We now proceed to illustrate the architecture and implementation of the Patient Observing and Monitoring MAS. A prototype instance of PoMAS, implemented in DALI, has, in fact, been experimented with in real settings in our past work.

PoMAS, as shown in Figure 2, comprises several agents; the two main ones are discussed below.

- *The SmartHealthHelper agent.*

The functions of this agent are listed below. Specifically, the agent:

- Analyzes the patient's vitals (e.g., heart rate, saturation, minimum and maximum pressure, temperature, weight, or whatever else is needed in each specific context) taken by wearable devices or by patients (or caregivers) themselves, returns immediate feedback on each parameter, and stores them in the corresponding list.
- Every time new parameters are entered, the agent compares them with the previously stored values, according to a sliding temporal window, to detect various potentially dangerous situations (such as tachycardia/bradycardia in case of heart rate, hypoxia in case of saturation, and hypertension/hypotension in case of pressure problems). Periodically, it invokes the Complex Event processing module on a section of the data stream recorded so far.



**Figure 2:** MAS Architecture

- Analyzes the symptoms that the patient reports.
- By examining the value of parameters and by exploiting the Complex Event Processing and the PredictMed modules, the agent identifies potentially dangerous situations. In this case, it contacts the *DoctorAgent*, which will provide feedback on what to do. Otherwise, it provides reassurance and behavioural suggestions, or minor re-arrangements of the dosage of drugs, that might alleviate the unpleasant sensations that the patient is experiencing.
- For each therapy entered by human doctors, i.e., a medicine to be taken regularly, sends a reminder to the user when necessary.
- Periodically, sends a reminder for temperature, blood pressure, or whatever measurements must be made by the patient or caregivers.
- Suggests immediate countermeasures for the patients if they feel unwell (e.g., in case of dizziness, high/low heart rate, etc.), pending medical intervention.
- With each new entry of a measurement value, compare it with the previous one and return feedback indicating possible necessary countermeasures to be taken even if the patient feels well (e.g., the blood pressure is high despite the patient not sensing it) and informs the *DoctorAgent*.

Vital parameters are, as mentioned, analyzed both over the short and over the long term by the Complex Event Processing Module. If a dangerous situation is suspected, the *SmartHealthHelper* will alert the *DoctorAgent*, which will be able to make a diagnosis and propose countermeasures based on the complex data and events obtained by *SmartHealthHelper*, and through the advice provided by a human doctor. All of the above is performed by the agent based on the patient's clinical history, and based on the following:

- ethical principles, both the general ones concerning the medical domain and the patient’s personal ones;
- patient’s preferences, concerning at least:
  - \* therapies that (s)he will accept or not;
  - \* degree and kind of information to be imparted to the patient or the caregivers;
  - \* medical specialists that the patient might prefer to consult;
  - \* other preferences, concerning, for instance, the time and money that the patient can spend for treatments or consultations.

- The *DoctorAgent*.

The functions of this agent are listed below. Specifically, the agent:

- Receives messages from the *SmartHealthHelper* agent, analyzes the communicated data and problems, e.g., fever or tachycardia, or, in our case, complex events detected via CEP, and responds by suggesting a medication to take or, in case of grave danger, it suggests going to the Emergency Room. It also provides feedback on when to take the next measurement of various kinds of data. Such feedback can be provided automatically and partly by asking a human doctor.
- Reports to the human medical doctors periodically and proactively in case of a suspected emergency.
- Is able, proactively or at the request of the human medical doctor, to provide data and invoke a Machine Learning module, in our case PredictMed, to estimate the actual probability of the presence of a dangerous event (e.g., referring to past work, serious heart events, or seizures). In the positive case, it informs the *SmartHealthHelper* agent so that it can tune its activities w.r.t. the reported illness, i.e., it can modify the set of vital signs and the symptoms to monitor and set specific danger conditions beyond alerting medical doctors and Emergency Room if needed.

Thus, in summary, the *smartHealthHelper* agent can detect alarming situations by comparing different measurements, such as pressure, over time, or by being directly informed by the patients (or caregivers) themselves of alarming symptoms. It can exploit Complex Event Processing. It resorts to the *DoctorAgent* if a potentially dangerous situation is detected. In the case of normal situations, clinical data are periodically delivered to the *DoctorAgent*. The *DoctorAgent* will return an outcome that is shown to the patient (or to the caregiver) via an app (so far developed for Android cell phones), which serves as a user interface and that can be installed on a cell phone or a PC. This outcome can be a piece of advice or suggestion, a prescription of new medicines to take/tests to be performed, or an alarm with the invitation to reach the nearest Emergency Room (as seen below, in future developments, it will directly send an ambulance/helicopter; this requires the proposed system to be integrated into the National Health System). Small suggestions (e.g., increasing the dosage of a medication) can be made on the agent’s own initiative, based on the patient’s medical records and the existing doctor’s prescriptions. Otherwise, interaction with human doctors is required. The *DoctorAgent* is in charge of interacting with human doctors and, upon their request, can exploit the PredictMed module.

In Figure 3 is a sample of the DALI code that detects symptoms related to heart conditions.

```

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 3:** A sample of the code used to detect possible health problems and alarming situations

### 3. An Healthcare-oriented Hybrid Society

Based on multi-agent and multi-MAS architectures proposed by some of the authors of this paper [29, 30, 31], PredictMed-PoMAS PMAs will be able to become components of a complex healthcare ecosystem, as discussed below, and even more at large, to a kind of *Hybrid Society* that we envisage.

#### 3.1. A Healthcare Ecosystem, some Examples of Functioning

The complex ecosystem we envisage is an open, dynamic computational environment that includes heterogeneous software components implemented as Multi-Agent Systems (MAS) modelling real-world entities. In healthcare, such a system may have objectives, e.g., monitoring and assisting patients, rationalizing access to medical doctors, ambulances, helicopters, and hospital beds, that they pursue autonomously, requiring human intervention only when needed. They should also be dynamic because they should not exploit not only a predefined set of knowledge bases and/or resources but also be able to locate new knowledge sources and reasoning services.

Below, we propose an example of what such a system might be able to do. Assume that, in

this system, each human patient is monitored by a PMA equipped with local 'contexts' (i.e., knowledge repositories), complex event processing and reasoning modules, and the PredictMed module. The contexts may provide the PMA with, e.g., the patient's clinical history, patient preferences and needs, information about the patient's standard treatment, and about possible actions to undertake in case of changes in some of the patient's parameters, for instance by rearranging the quantity of medicine according to specific values in a blood test. The CEP modules and the PredictMed module can, as seen before, detect symptoms and decide whether they correspond to a potentially serious or unexpected situation. A reasoning module might devise a plan for coping with such situations.

The system will encompass several PMAs in charge of different patients and quite complex MASs, modelling, for instance, hospitals, or one or more "Diagnostic Centers" providing more in-depth interpretation of symptoms, "Medical Centers" providing consultation with human specialists, and "Emergency Centers" managing hospital beds and transportation facilities. In our architecture, all these complex MASs are supposed to encompass an *Institutional Agent* in charge of interfacing with external and internal entities. Single users, such as medical specialists, participate through their Personal Assistant Agent (PA). PAs, in fact, of which PMAs can be seen as a particular case devised for patients, will represent the human users' entry point into the Hybrid Society.

Institutional Agents can locate (the Institutional Agents or the PAs of) specific components within the system, searching through their names and/or roles.

Below, we provide examples of how each PMA can proactively resort to such external systems. Committed to Computational Logic, we employ a syntax reminiscent of logic programming. This syntax (yet to be implemented) gracefully extends DALI syntax.

Consider, for instance, the rule below:

$$G\langle 8h \rangle \text{ high\_blood\_pressure } \mathbf{enables} \\ \text{communication}(pma, \text{helpdesk}@Inst@medcenter, \\ \text{cardiological\_consultation\_required}(patient_{pma}, \text{high\_blood\_pressure}))$$

Here,  $G$  stands for "always" (with the usual meaning borrowed from LTL temporal logic [32]) where  $\langle 8h \rangle$  indicates an interval including the last 8 hours. So, the patient in charge of a certain PMA has had high blood pressure for the last eight hours. This enables the PMA to undertake suitable countermeasures. Here, the PMA communicates with the agent in charge of dispatching consultation requests to an available Medical Center, identified by the expression  $\text{helpdesk}@Inst@medcenter$  (where  $Inst$  denotes the Medical Center's institutional agent), to require a cardiological consultation due to lasting high blood pressure. Precisely, the reference to a suitable helpdesk agent  $\text{helpdesk}$  at an available Medical Center  $\text{medcenter}$  is provided by the centre's Institutional Agent  $Inst$ , inquired by role.

In the following rule, a PMA in charge of an emergency is enabled to require urgent transportation of the patient in its charge to the hospital because of emergency condition  $E$ . The request is issued to the manager agent of the Emergency Center MAS, identified by role via the expression  $\text{emergency\_manager}@Inst@emergencies$ . Thus:

*emergency(E)* enables  
*communication(pma, emergency\_manager@Inst@emergencies,*  
*hospital\_transportation\_required(patient<sub>pma</sub>, condition(E)))*

There can also be, for instance, the option of looking for a specialist from a medical directory, represented, e.g., by a component called *med-dir*. A patient's PMA might use a rule such as:

$$find\_cardiologist(N) \leftarrow med-dir : cardiologist(N)$$

where the query to the medical directory will return in variable *N* the name (e.g., Maggie Smith) of a cardiologist. Actually, the rule will return the name of the cardiologist's PA, in charge of making appointments.

After acquiring, as seen above, the reference to a reliable cardiologist (i.e., *maggie-smith*), the patient (say Mary) can get in contact with the cardiologist, disclose her health condition *C*, and thus make an appointment for time *T*. This, in our approach, can be made by a rule like:

$$make\_appointment(mary, T) \leftarrow$$

$$condition(mary, C),$$

$$maggie-smith : consultation\_needed(mary, C, T)$$

We may notice that the "commitments" component, not discussed in this paper, plays a fundamental role. The agents receiving a request will commit to satisfy such request in a certain way and within a specific time: e.g., the Medical Center will provide a video conference with Dr. House, and the Emergency Center will commit to sending, e.g., an ambulance by the hour or if deemed necessary, a helicopter by twenty minutes.

### 3.2. The Hybrid Society

In the above examples, the involved sub-systems may have different objectives: a PMA has the objective to keep the health of its assigned patient under control. The emergency centre will have the objective of optimal or quasi-optimal usage of the available resources, such as doctors, ambulances, helicopters, etc. The overall system will aim to devise a suitable compromise, i.e., caring best for patients' health while staying within budget limits and resource availability. For this planning activity, we intend to leverage the potential of Answer Set Programming (ASP) in future work. ASP is a successful logic programming paradigm (cf. [33] and the references therein). ASP is implemented by employing effective inference engines, called *solvers*, where many performant ASP solvers [34] are available as open-source tools. ASP has been widely applied in many fields, e.g., information integration, constraint satisfaction, routing, planning, diagnosis, configuration, computer-aided verification, biology/biomedicine, knowledge management, etc.

Notice also that ontologies [35, 36] and ontological reasoning are helpful in such a context to describe, e.g., resources, patients, diseases, etc.

A system such as the one that we outlined can be seen as an instance of what we may call a "Hybrid Society" (HS), where several kinds of Autonomous Systems (AS, among which, in our case, the PMAs, the PAs, the MASs with their Institutional Agents) and human users can be

involved. The Hybrid Society is the theme of the PRIN Project *TrustPACTX - Design of the Hybrid Society Humans-Autonomous Systems: Architecture, Trustworthiness, Trust, EthiCs*, involving the Universities of L'Aquila (Stefania Costantini, PI), Messina (Pasquale De Meo), Napoli Federico II (Silvia Rossi), and the National Research Council (Rino Falcone).

The approach proposed in the project for the development of the Hybrid Society, especially for applications as crucial as healthcare, is based on three main *pillars*:

1. Verifiability (trustworthiness). This notion involves mechanisms to ensure that all components of the hybrid society are compliant concerning their expected behaviour and, in case of violations, suitable measures will be enacted, independently of the size and complexity of the system
2. Trust and Ethics. An autonomous collaborative system is based on trust in its actions and decisions and must obey ethical principles, including general and user-specific ones. In healthcare, for instance, as for general principles, the physician has an ethical obligation (i) to benefit the patient, (ii) to avoid or minimize harm, and (iii) to respect the values and preferences of the patient, principles that must be then much better detailed, entailing for instance that medical interventions cannot be performed without informed consent. As for user-specific ones, for instance, some minority populations hold views different from those of the majority about the need for full disclosure and decisions about life support. Trust must be measured concerning the dynamics that determine the behaviour of the autonomous system: its specific operational capabilities, deontic capacity, and the control system put in place. The user's perception of the level of trust and faith in the ethical behaviour attributable to the PA impacts the motivations and intentions of humans to use and embrace such a system.
3. Explainability. What drives decisions of AS are, initially, the goals instilled by system designers and, later on, beliefs, intentions, and goals developed during their operation (according to the well-known BDI logic model, to which languages to program such systems are usually inspired). Therefore, their behaviour and decisions may be hard to understand for humans. Human users would (rationally) trust more those AS that could explain their behaviours and choices intelligibly. Numerous studies have linked trust to the possibility of having a verifiable system that can provide explanations that are understandable by each specific category of user.

In the HS, humans and autonomous systems (here, the PMAs and the representative agents of the various participating doctors and institutions) should be coupled at multiple levels based on shared agreed-upon principles and standards, which must, by definition, enforce tight constraints on agents' behaviour. Such principles should include values, social norms, and ethical and professional conduct codes relative to healthcare. They should be flexible and capable of evolving over time according to changes and evolution in context, needs, or norms (we refer to the High-Level Expert Group on Artificial Intelligence. 'Ethics Guidelines For Trustworthy AI'. Brussels: European Commission, 2019, [https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/horizon/guidance/ethical-guidance-for-research-with-a-potential-for-human-enhancement-sienna\\_he\\_en.pdf](https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/horizon/guidance/ethical-guidance-for-research-with-a-potential-for-human-enhancement-sienna_he_en.pdf)).

For all the three pillars above, adequate levels of accountability and traceability are needed in the design, development, and deployment of PMAs, PAs, and Institutional Agents. Accountability

and traceability duties find cornerstones in several regulations (e.g. the GDPR, in terms of accountability and paths to balance the protection of fundamental rights and the pursuit of innovation) and in the high standard of care by developers and researchers (regarding national and international standards). Concerning the definitions of actors in the Hybrid Society and the responsibilities they must take therein, we will consider recent interdisciplinary research [37, 38]. It is unclear whether and to what extent artificial entities can be responsible in the same sense as human beings. However, based on the conceptual clarification, work can be undertaken to develop a concept of responsibility in Hybrid Societies, which is one of the objectives of TrustPACTX.

## 4. Conclusion

In this paper, we abstracted from our past work to outline an architecture for the monitoring and assistance of patients, featuring PMAs (Personal Monitoring Agents) based upon a specialized Multi-Agent System, PoMAS, encompassing both a Complex Event Programming ISEQL component and a (trained) Machine Learning component (Predicted). Two instances of the architecture have been prototypically implemented in past work to monitor older adults with heart issues and fragile children.

We have then outlined our work in progress, which aims to consider PMAs, which will be instances of our architecture as components of a wider architecture encompassing many components of the healthcare systems, represented by PAs and Institutional Agents. This wider architecture can be seen as an instance of an envisaged Hybrid Society that we intend to study and develop, with particular attention to the three pillars of verifiability, trust and ethics, and explainability that we deem indispensable in such a landscape.

This work and the TrustPACTX project have the ultimate objective of modernizing and *humanizing* the National Health System, making it more robust in facing long-term challenges such as providing high-quality healthcare for everyone in a landscape of limited resources due to the many needs of an ageing population. This may, in perspective, deeply impact the digital transformation of the health system, which is one of the core goals of the Next-EU program ([https://ec.europa.eu/health/funding/eu4health\\_en](https://ec.europa.eu/health/funding/eu4health_en)).

## References

- [1] L. D. Lauretis, F. Persia, S. Costantini, D. D’Auria, How to leverage intelligent agents and complex event processing to improve patient monitoring, *J. Log. Comput.* 33 (2023) 900–935. URL: <https://doi.org/10.1093/logcom/exad016>. doi:10.1093/logcom/exad016.
- [2] C. M. Bertonecchi, S. Costantini, F. Persia, D. Bertonecchi, D. D’Auria, Predictmed-epilepsy: A multi-agent based system for epilepsy detection and prediction in neuropsychiatry, *Comput. Methods Programs Biomed.* 236 (2023) 107548. doi:10.1016/j.cmpb.2023.107548.
- [3] L. D. Lauretis, S. Costantini, E. Pallotta, C. Balsano, An ontology of medical wearables, in: *IEEE International Conference on Consumer Electronics, ICCE 2022, Las Vegas, NV, USA, January 7-9, 2022*, IEEE, 2022, pp. 1–4. doi:10.1109/ICCE53296.2022.9730210.

- [4] C. Bertongcelli, P. Altamura, E. Vieira, D. Bertongcelli, S. Thummler, F. Solla, Identifying factors associated with severe intellectual disabilities in teenagers with cerebral palsy using a predictive learning model, *J. of Child Neurol.* (2019). doi:10.1177/0883073818822358.
- [5] C. Bertongcelli, P. Altamura, E. Vieirs, D. Bertongcelli, F. Solla, Using artificial intelligence to identify factors associated with autism spectrum disorder in adolescents with cerebral palsy, *Neuropediatrics* 50 (2019) 178–187. doi:10.1055/s-0039-1685525.
- [6] C. Bertongcelli, P. Altamura, E. V. E.R., S. Iyengar, F. Solla, D. Bertongcelli, Predictmed: A logistic regression-based model to predict health conditions in cerebral palsy, *Health Informatics J.* 26 (2020). doi:10.1177/1460458219898568.
- [7] C. Bertongcelli, P. Altamura, S. Bagui, et al, Predicting osteoarthritis in adults using statistical data mining and machine learning, *Therapeutic Advances in Musculoskeletal Disease* (2022). doi:https://doi.org/10.1177/1759720X221104935.
- [8] H. Hamann, Y. Khaluf, J. Botev, M. D. Soorati, E. Ferrante, O. Kosak, J. Montanier, S. Mostaghim, R. Redpath, J. Timmis, F. Veenstra, M. Wahby, A. Zamuda, Hybrid societies: Challenges and perspectives in the design of collective behavior in self-organizing systems, *Frontiers Robotics AI* 3 (2016) 14. doi:10.3389/frobt.2016.00014.
- [9] U. Thomas, Proceedings The first international Conference on Hybrid Societies, Chemnitz University of Technology, 2023. Forthcoming.
- [10] A. Poggi, M. Tomaiuolo, P. Turci, A testbed for agent mediated service composition, in: Proc. of the Fourth International Symposium "From Agent Theory to Agent Implementation" (AT2AI-4), volume 2, 2004, pp. 529–534.
- [11] A. Moreno, D. Isern, D. Sánchez, Provision of agent-based health care services, *AI Commun.* 16 (2003) 167–178.
- [12] R. Calegari, G. Ciatto, V. Mascardi, A. Omicini, Logic-based technologies for multi-agent systems: a systematic literature review, *Auton. Agents Multi Agent Syst.* 35 (2021) 1. doi:10.1007/s10458-020-09478-3.
- [13] M. Fisher, V. Mascardi, K. Y. Rozier, B. Schlingloff, M. Winikoff, N. Yorke-Smith, Towards a framework for certification of reliable autonomous systems, *Auton. Agents Multi Agent Syst.* 35 (2021) 8. doi:10.1007/s10458-020-09487-2.
- [14] S. Costantini, Ensuring trustworthy and ethical behaviour in intelligent logical agents, *J. Log. Comput.* 32 (2022) 443–478. doi:10.1093/logcom/exab091.
- [15] A. Dyoub, S. Costantini, F. A. Lisi, I. Letteri, Logic-based machine learning for transparent ethical agents, in: F. Calimeri, S. Perri, E. Zumpano (Eds.), Proceedings of the 35th Italian Conference on Computational Logic - CILC 2020, Rende, Italy, October 13-15, 2020, volume 2710 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 169–183.
- [16] A. Dyoub, S. Costantini, I. Letteri, Care robots learning rules of ethical behavior under the supervision of an ethical teacher (short paper), in: P. Bruno, F. Calimeri, F. Cauteruccio, M. Maratea, G. Terracina, M. Vallati (Eds.), Joint Proceedings of the 1st International Workshop on HYbrid Models for Coupling Deductive and Inductive ReAsoning (HYDRA 2022) and the 29th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (RCRA 2022) co-located with the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022), Genova Nervi, Italy, September 5, 2022, volume 3281 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 1–8.

- [17] S. Helmer, F. Persia, High-level surveillance event detection using an interval-based query language, in: Tenth IEEE International Conference on Semantic Computing, ICSC 2016, Laguna Hills, CA, USA, February 4-6, 2016, IEEE Computer Society, 2016, pp. 39–46. doi:10.1109/ICSC.2016.19.
- [18] S. Helmer, F. Persia, Iseq, an interval-based surveillance event query language, *Int. J. Multim. Data Eng. Manag.* 7 (2016) 1–21. doi:10.4018/IJMDEM.2016100101.
- [19] 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.
- [20] M. Baldoni, F. Bergenti, A. E. F. Seghrouchni, M. Winikoff, Special issue on current trends in research on software agents and agent-based software systems (guest editorial), *Auton. Agents and Multi-Agent Syst.* 35 (2021). doi:s10458-021-09510-0.
- [21] S. Costantini, A. Tocchio, A logic programming language for multi-agent systems, in: S. Flesca, S. Greco, N. Leone, G. Ianni (Eds.), *Logics in Artificial Intelligence, European Conference, JELIA 2002, Proceedings, volume 2424 of Lecture Notes in Computer Science*, Springer, 2002.
- [22] S. Costantini, A. Tocchio, The DALI logic programming agent-oriented language, in: J. J. Alferes, J. A. Leite (Eds.), *Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, Proceedings, volume 3229 of Lecture Notes in Computer Science*, Springer, 2004, pp. 685–688.
- [23] G. De Gasperis, S. Costantini, G. Nazzicone, Dali multi agent systems framework, doi 10.5281/zenodo.11042, DALI GitHub Software Repository, 2014. DALI: <http://github.com/AAAI-DISIM-UnivAQ/DALI>.
- [24] S. Costantini, A. Tocchio, A. Verticchio, Communication and trust in the DALI logic programming agent-oriented language, *Intelligenza Artificiale* 2 (2005) 39–46. *Journal of the Italian Association AI\*IA*.
- [25] S. Costantini, G. D. Gasperis, V. Pitoni, A. Salutari, DALI: A multi agent system framework for the web, cognitive robotic and complex event processing, in: D. D. Monica, A. Murano, S. Rubin, L. Sauro (Eds.), *Joint Proceedings of the 18th Italian Conference on Theoretical Computer Science and the 32nd Italian Conference on Computational Logic co-located with the 2017 IEEE International Workshop on Measurements and Networking (2017 IEEE M&N)*, Naples, Italy, September 26-28, 2017, volume 1949 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 286–300. URL: <https://ceur-ws.org/Vol-1949/CILCpaper05.pdf>.
- [26] S. Costantini, G. De Gasperis, G. Nazzicone, DALI for cognitive robotics: Principles and prototype implementation, in: Y. Lierler, W. Taha (Eds.), *Practical Aspects of Declarative Languages - 19th International Symposium, Proceedings, volume 10137 of Lecture Notes in Computer Science*, Springer, 2017, pp. 152–162.
- [27] S. Costantini, G. De Gasperis, V. Pitoni, A. Salutari, Dali: A multi agent system framework for the web, cognitive robotic and complex event processing, in: *Proceedings of the 32nd Italian Conference on Computational Logic*, volume 1949 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 286–300. [Http://ceur-ws.org/Vol-1949/CILCpaper05.pdf](http://ceur-ws.org/Vol-1949/CILCpaper05.pdf).
- [28] S. Costantini, G. D. Gasperis, A. Provetti, P. Tsintza, A heuristic approach to proposal-based negotiation: With applications in fashion supply chain management, *Mathematical Problems in Engineering* (2013). Article number 896312.

- [29] S. Costantini, Knowledge acquisition via non-monotonic reasoning in distributed heterogeneous environments, in: F. Calimeri, G. Ianni, M. Truszczyński (Eds.), *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015*, Lexington, KY, USA, September 27-30, 2015. Proceedings, volume 9345 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 228–241. URL: [https://doi.org/10.1007/978-3-319-23264-5\\_20](https://doi.org/10.1007/978-3-319-23264-5_20). doi:10.1007/978-3-319-23264-5\_20.
- [30] S. Costantini, V. Pitoni, K-ACE: A flexible environment for knowledge-aware multi-agent systems, in: M. Baldoni, M. Dastani, B. Liao, Y. Sakurai, R. Zalila-Wenkstern (Eds.), *PRIMA 2019: Principles and Practice of Multi-Agent Systems - 22nd International Conference*, Turin, Italy, October 28-31, 2019, Proceedings, volume 11873 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 19–35. URL: [https://doi.org/10.1007/978-3-030-33792-6\\_2](https://doi.org/10.1007/978-3-030-33792-6_2). doi:10.1007/978-3-030-33792-6\_2.
- [31] P. Cabalar, S. Costantini, G. D. Gasperis, A. Formisano, Multi-context systems in dynamic environments, *Ann. Math. Artif. Intell.* 86 (2019) 87–120. URL: <https://doi.org/10.1007/s10472-019-09622-0>. doi:10.1007/s10472-019-09622-0.
- [32] D. Gabbay, C. Hogger, J. Robinson, (eds.), volume 4 (epistemic and temporal reasoning), in: *Handbook of Logic in AI and Logic Programming*, Oxford University Press, 1995.
- [33] G. Brewka, T. Eiter, M. T. (eds.), Answer set programming: Special issue, *AI Magazine* 37 (2016).
- [34] M. Gebser, N. Leone, M. Maratea, S. Perri, F. Ricca, T. Schaub, Evaluation techniques and systems for answer set programming: a survey, in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization*, 2018, pp. 5450–5456. URL: <https://doi.org/10.24963/ijcai.2018/769>. doi:10.24963/ijcai.2018/769.
- [35] M. T. Schneider, Ontologies and data management: A brief survey, *Künstl Intell.* 34 (2020) 329–353. doi:10.1007/s13218-020-00686-3.
- [36] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, *The descriptive logic handbook: Theory, implementation, and applications*, Cambridge Univ. Press, 2003.
- [37] S. Meyer, S. Mandl, Gesmann-Nuissl, E. al., Responsibility in hybrid societies: concepts and terms, *AI Ethics* 3 (2023) 25–48. doi:10.1007/s43681-022-00184-2.
- [38] R. Falcone, A. Sapienza, The role of decisional autonomy in user-iot systems interaction, in: A. Ferrando, V. Mascardi (Eds.), *Proceedings of the 23rd Workshop "From Objects to Agents"*, Genova, Italy, September 1-3, 2022, volume 3261 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 77–87.