J Med Syst manuscript No.
(will be inserted by the editor)

Processing Diabetes Mellitus Composite Events in MAGPIE

Albert Brugués · Stefano Bromuri · Michael Barry · Óscar Jiménez del Toro · Maciej R. Mazurkiewicz · Przemyslaw Kardas · Josep Pegueroles · Michael Schumacher

Received: date / Accepted: date

Abstract
Research focus: the focus of this research is in the definition of programmable expert personal health systems to monitor patients affected by chronic diseases using agent oriented programming and mobile computing to represent the interaction happening amongst the components of the system. The paper also discusses issues of knowledge representation within the medical domain when dealing with temporal patterns concerning the physiological values of the patient.

Research method: We evaluate the presented agent based PHS against its scalability, by comparing it with a centralized approach and then we also use the data of 21 diabetic patients to evaluate the accuracy of a set of dynamic monitoring rules defined.

Results: The evaluation concerning the scalability of the system illustrates the fact that a centralized approach towards monitoring chronic illnesses is not scalable and an approach making use of mobile computing and agents experts is more likely to satisfy the needs of next generation PHSs. The evaluation also highlights the advantages of having dynamic rules to monitor patients, evaluating the precision of three of such rules.

Conclusions: PHSs are becoming an adopted technology to deal with the surge of patients affected by chronic illnesses. In this paper we discussed architectural choices to make an agent based PHS more scalable by using a distributed mobile computing approach. We also discussed how to model the medical knowledge in the PHS in such a way that this is modifiable at run time. The evaluation highlights the necessity of distributing the reasoning to the mobile part of the system and the necessity to define modifiable rules to be able to deal with the change in lifestyle of the patients affected by chronic illnesses.

Keywords Pervasive healthcare · Agents · Distributed expert systems · Event calculus · Scalability · Diabetes mellitus

1 Introduction

Diabetes mellitus (DM) is a chronic health condition in which the pancreas does not produce enough insulin, or the body cannot use it effectively. The former is known as DM Type I and patients belonging to this group need external administration of insulin, the latter is DM Type II and does not usually require external administration of insulin. A third type is Gestational DM, which can appear during the pregnancy. In all cases the complications may lead to have episodes with high blood glucose levels (hyperglycemia) or low blood glucose levels (hypoglycemia). A person with DM can develop long-term complications such as damage to small blood vessels in the kidneys (nephropathy) or in the eyes (retinopathy), damage to nerves (neuropathy), and twice the risk of having cardiovascular disease [10]. The prevention and treatment of DM includes a healthy diet, a regular physical activity, maintaining a normal

Albert Brugués (🖂) · Stefano Bromuri · Michael Barry · Óscar Jiménez del Toro · Michael Schumacher
University of Applied Sciences
Western Switzerland (HES-SO),
Techno-Pôle 3, CH-3960 Sierre, Switzerland
E-mail: albert.brugues@hevs.ch

Albert Brugués · Josep Pegueroles
Universitat Politècnica de Catalunya - BarcelonaTech (UPC),
Campus Nord, Edif. C3, C. Jordi Girona, 1-3,
08034 Barcelona, Spain

Maciej R. Mazurkiewicz · Przemyslaw Kardas
First Department of Family Medicine
Medical University of Lodz
ul.Narutowicza 60, Łódź

body weight, and not smoking. It is estimated that DM will affect 439 million adults by 2030 [34], and that health expenditures for DM will be USD 490 billion by the same year [46].

The use of pervasive healthcare [4] is a prominent way to reduce healthcare costs caused by the prevalence of chronic diseases like DM. Pervasive healthcare implies the decentralization of healthcare services by focusing on the patients rather than the doctors [2], and removing physical and time barriers in healthcare to enable the paradigm of *"healthcare to anyone, anytime and anywhere"* [40]. All these goals are achieved with the definition of Personal Health Systems (PHSs). The typical architecture of a PHS consists on three tiers [38] namely: Tier 1 Body Area Network (BAN), Tier 2 base station and Tier 3 remote monitoring server. The BAN consists in a set of sensors deployed on the body of the patient for monitoring physiological parameters. The base station can be a tablet, a mobile phone or a portable device which collects and aggregates the health data produced by the BAN and sends them to the remote server over the Internet. The remote server provides assistance to patients and helps doctors on the management of their patients, and at the same time saves the patient's data in the hospital's database for electronic health records.

There is a wide range of applications for PHSs, some provide passive safety like requesting help to caregivers in case of a fall event [1], while others are focused on the prevention of diseases like depression [31]. In all cases, the development of such systems have some technological challenges in common that must be taken into account. Amongst them are the modeling of the medical knowledge, the scalability of the system, the personalization of healthcare services for each patient and the interoperability between different heterogeneous systems.

In the context of PHSs the medical knowledge must be transformed to provide clinical decision support, and improve the healthcare safety and assistance offered to the patient. When there are data available from the domain of interest, the typical approach is to use machine learning techniques to train classifiers to assist in that particular domain. Moreover, these classifiers can be customized for each patient in order to offer the next generation of healthcare services [11]. Unfortunately, the issue of defining a classifier becomes quite difficult when dealing with temporal patterns that may have an irregular number of events as features. In this sense, in this paper we study the definition of rules that can be modified and personalized according to the preference of medical doctors, to deal with patterns of this kind.

Another desired feature for providing good healthcare services with PHSs is scalability, that is the performance of these systems must not degrade when the number of patients increases, and specially to handle a big-data scenario where the number of patients can be as large as an entire city being everyone monitored [26]. Last, it is desired that different systems are interoperable so that they are able to *"exchange information and use the information that has been exchanged"* [20]. It has been reported that current systems are closed in nature and therefore not supporting a collaborative behavior [5].

This paper shows a PHSs for DM management with the aim of minimizing the risk of developing the health complications related with this disease. The design of this PHS takes into consideration the previously explained issues.

The main component of this system is the MAGPIE agent platform [8], which has been designed to run in Android handheld devices. The use of agents in PHSs can simplify the modeling of medical knowledge as they are autonomous software entities, that pursue a set of goals [44] in an intelligent way, by applying Artificial Intelligent reasoning techniques such as deduction, and act proactively, without necessarily receiving a stimulus from the user. This set of properties can benefit the current definition of PHSs, by having monitoring tools that are capable of reasoning in a complex and proactive way on the current patients' physiological parameters. Moreover, the deployment of the agents in the Tier 2, that is in the mobile device from each patient, improves the scalability of the PHS in comparison with the current state-of-the-art approach where the computations for patient monitoring are done in Tier 3. This last tier is shared by all the patients using a particular PHS and therefore represents an inherent bottleneck of the system.

Concerning the personalization of healthcare services, medical doctors using the PHS can define monitoring rules for each patient using a graphical web interface. The creation of these rules is based on the combination of different events related to the monitoring of the patient, and the definition of the thresholds values that make this events happen. This is an important feature as a person can be considered to have the blood pressure high if it is 140/90 mmHg, but this measurement can indicate an improvement if the person had higher values in the past.

Finally, to achieve interoperability with other medical systems the CDA standard [14] is used. In particular, the information flowing from Tier 2 to Tier 3 is encoded according to this standard. The details on this part are out of the scope of this paper and the interested reader can find more information in [9].

2 System overview

The PHS for DM monitoring reported in this paper has two different actors: patients and doctors (see Figure 1). The medical side of the system corresponds to the Tier 3 of a PHS. In this case, healthcare professionals can interact with the system by means of a web application that allows them to visualize and analyze data from patients, and to manage and define specific monitoring rules for each of the patients. Regarding the patient side it maps to the Tier 1 and Tier 2, where patients with DM are monitored by means of using an Android smartphone (Tier 2) and a set of sensors that conforms the BAN (Tier 1).

The smartphone has installed a mobile application with the purpose of managing the DM. This application is developed with the MAGPIE framework, so its based on a multiagent system that is able to perceive the physiological values measured by the sensors. The agents from that system are responsible to perform reasoning on these data and to produce alerts according to the monitoring rules defined by the doctors. This design strategy, where the computations are done in Tier 2 rather than in Tier 3, is expected to improve the scalability of the system, as Tier 3 is a component shared by all the patients of the system and can consequently become a bottleneck.

The evaluation of MAGPIE has been done using data collected with sensors from the COMMODITY$_{12}$ project [22]. This project consists of a PHS for the monitoring of patients affected with DM, where the following sensors are used in the BAN:

- GlucoTel [1] for capillary blood glucose measurements in mmol/l made six times per day on Mondays, before and after each meal; and one measurement during the morning the rest of the days.
- PressureTel [1] for the measurement of the blood pressure in mm Hg twice a day; one measurement during the morning and one during the evening.
- WeightTel [1] a scale used to measure the weight of the patient in kg once a day.

2.1 MAGPIE agent platform

MAGPIE is an agent platform integrated with the Android OS [8]. It plays the role of Tier 2 in a PHS by connecting the patient and the doctor, aiming to improve the management of chronic diseases. From the side of the patient it collects the physiological values measured by the sensors of the BAN, whereas from the medical side it contains the medical expertise provided by the doctor. By analyzing the physiological values according to the medical expertise, the agents of MAGPIE are able to produce alerts when there is an event of interest related to the illness being monitored.

MAGPIE is based on the concept of environment as a first class abstraction proposed by Weyns et al. [43]. This means that the environment should be considered as an implicit part of multiagent systems, mediating the interaction between agents and the access to resources. In MAGPIE, there is a relation with the virtual environment of a multiagent system and the real environment of a chronic patient, so that there is a data flow between both environments.

The MAGPIE agent platform consists of different components. The central element of the platform is the *environment* where it can be deployed with two main entities: *agents* and *context entities*. Agents are cognitive entities deployed on the agent environment. They share a similar architecture with agents of other platforms like PROSOCS [35], and GOLEM [7]. As in these platforms agents are composed of a declarative mind called agent mind and a body. The mind is the component in charge of the agent's reasoning abilities, and it is situated in the environment through the component called agent body. The agent body is the part of the agent that receives and produces events from/to the agent environment, so it acts as an interface between the agent mind and the agent environment.

Context entities are the connectors linking the real environment with the agent environment. They encapsulate the communication with a source of information from the real world. The goal of a context entity is to throw to the agent environment events related with physical measurements from the real world, so that the agents can perceive them. A context entity can also communicate events happening in the agent environment to the external world, such as alerts produced by the agents. There are different kinds of context entities modeling different sources of information in a PHS scenario. First, measurements can come from the Bluetooth sensors conforming the BAN of the patient, which can measure physiological values like the heart rate. Second, measurements can come from the sensors of the smartphone, which, for example, can provide the GPS position of the patient. Third, measurements can be provided by the patient itself through the user interface of a mobile application to report values that are difficult to measure with sensors, such as the amount

[1] http://www.bodytel.com

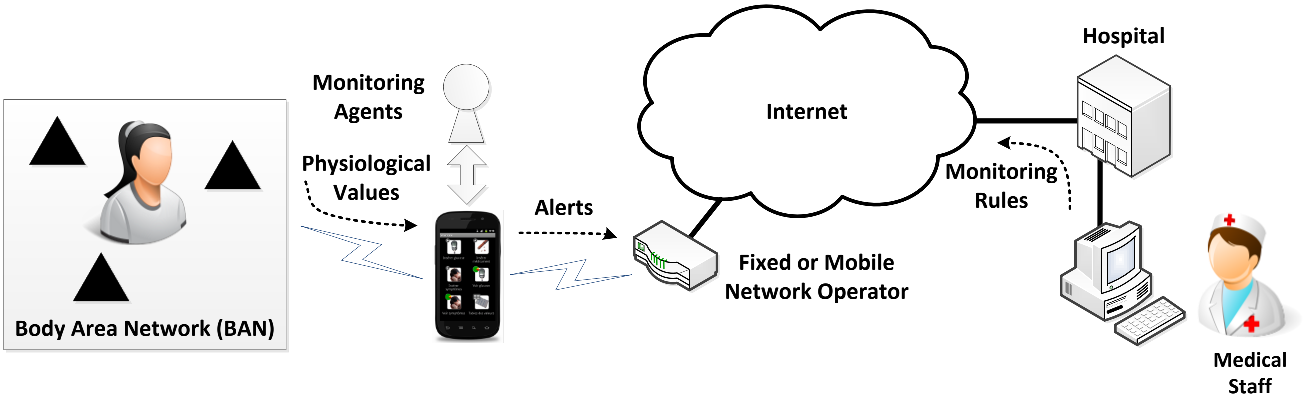
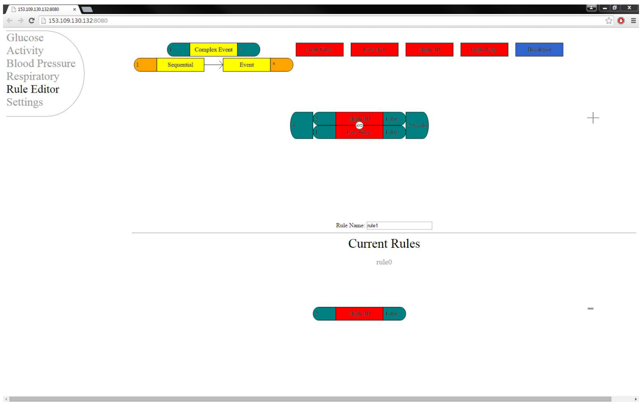


Fig. 1 Architecture of a PHS developed with MAGPIE



Rule Name: rule1

Current Rules

rule0

Fig. 2 Screenshot of the application for creating monitoring rules

of carbohydrates of a meal. Last, monitoring rules and alerts can be received or sent respectively to an external remote monitoring server.

2.2 Web interface for monitoring rules

The agents of MAGPIE must provide alerts according to monitoring rules, and medical doctors are the users of the system who have the knowledge to define such monitoring rules. However, they may not have knowledge on logic programming to define specific monitoring rules that are understandable by agents. To help medical doctors to play their role in an independent way, one of the components of the PHS is a web interface where doctors can program monitoring rules for the agents in a graphical way. Figure 2 shows how this application looks like.

In the context of the reported PHS a monitoring rule is defined as a combination of events that trigger an alert to be notified to a medical doctor; where an event is considered as the measurement of a physiological parameter categorized as high, normal or low. The physiological parameters considered for the implementation of the web interface are the ones measured in the COMMODITY$_{12}$ project, which are: glucose, blood pressure and weight. By combining these events two kinds of monitoring rules can be defined:

- Complex rules: involve the combination of two or more events in a given time window, where it is not considered the order in which the events happen.
- Sequential rules: involve the sequence of two or more events in a given time window, where the particular order in which the events occur matters.

Moreover, to provide personalized healthcare services, the medical doctor can define the high and low thresholds for each physiological parameter and patient, and therefore the normal range too. The creation of a monitoring rule follows the approach of a visual programming language, where different graphical elements are combined together to define the logic of a computer program. As stated before, monitoring rules are a combination of events, and therefore this is the only kind of graphical element that must be modeled, which in turn minimizes the complexity for creating the rules. Figure 3 shows the graphical design of an event, where *Event* defines the category and the name of a physiological parameter (i.e. high blood pressure); T is the time window given to the event, that is the amount of time after the specified event happens; and N is the number of times that the event must repeat for the given time window.

To create a monitoring rule the doctor must first select the type of rule, and then he can drag and drop the graphical elements representing the events that must happen in order to trigger the alert. The events are matched together vertically for complex rules and hori-

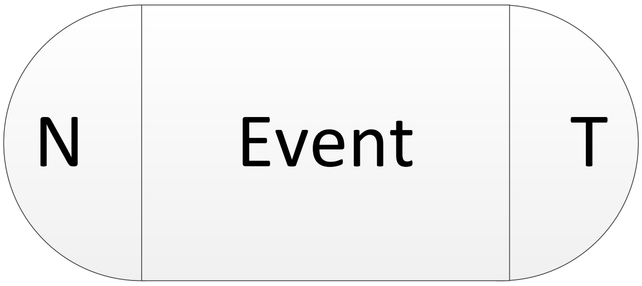


Fig. 3 Graphical representation of an event

zontally for sequential rules. Last, he can configure the parameters defining the event, and the message that must be displayed if the alert is triggered.

Figure 4 shows the relation between the graphical representation and the temporal representation of different rules that can be build with the web interface. Figure 4(a) shows the simplest rule that can be build, which is given as an illustrative example. This rule states that an alert is triggered if there is a single event of high blood pressure at time t_0. However, the web interface is intended to build complex and sequential monitoring rules involving several events. Figure 4(b) shows an example of a monitoring rule being complex and sequential at the same time, as it involves just one physiological parameter with a particular category. This rule states that an alert is triggered if the patient has two measurements of high blood pressure in a time period of one day. Figure 4(c) shows an example of a complex rule involving three events with two physiological parameters. This rule states that an alert is triggered when there are two events of high blood pressure and one event of low glucose in a time period of one day, where the time begins counting when the first of the events happens. The web interface also allows the user to create meta-rules. As shown in Figure 4(d) a meta-rule is a rule composed with a rule. In this case an alert is triggered if the pattern of events defined by the previous rule repeats three times in a period of two weeks.

2.3 Interactions between the different components

The management of a chronic illness like DM involves the implication of at least two individuals: the patient itself and the doctor in charge of the patient. When the management is done through a PHS developed with MAGPIE (Figure 1) there are also technological components involved like agents, the agent environment, sensors and web services. This section shows how patients, doctors and these components interact in order to improve the patients' quality of life.

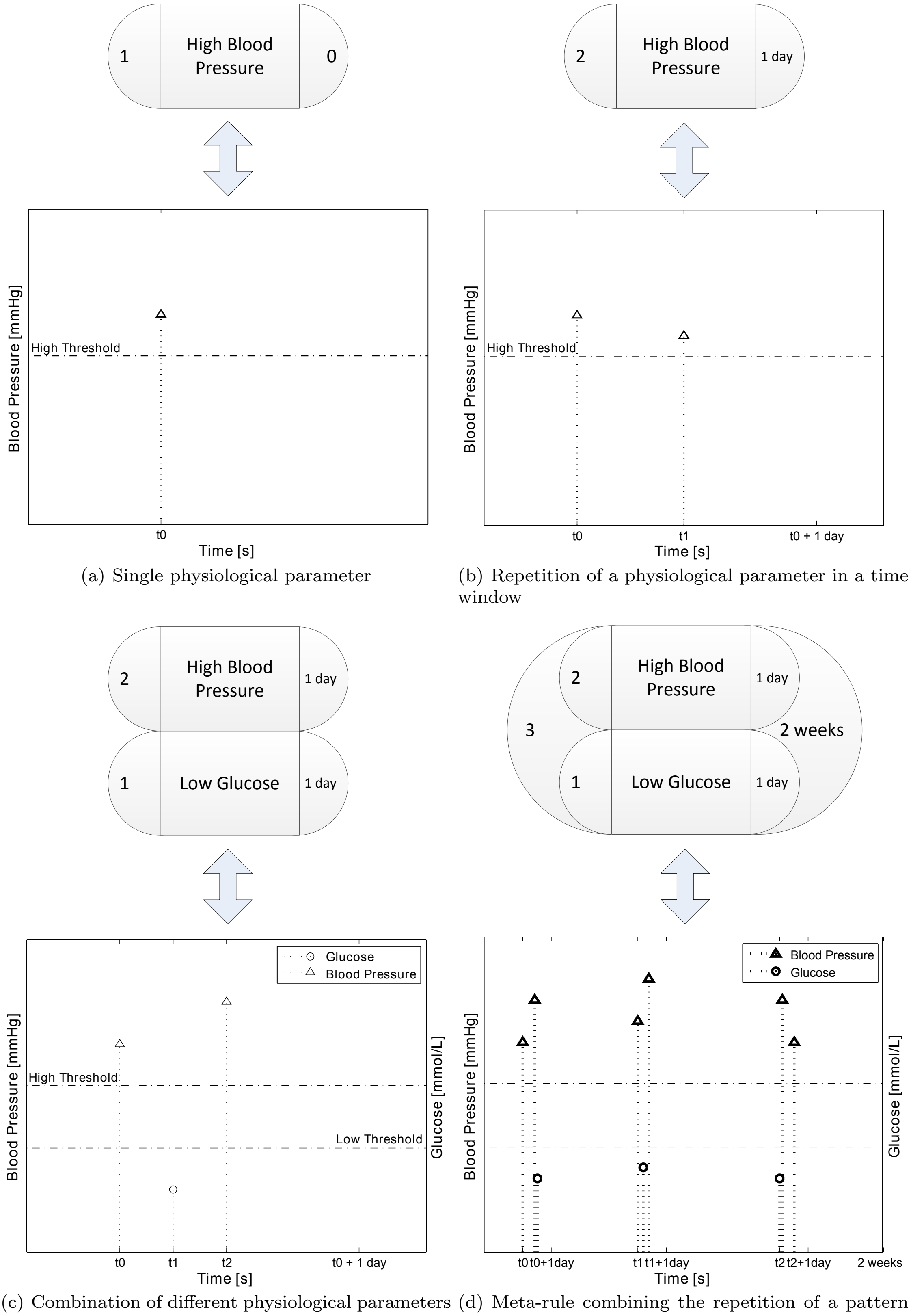
2.3.1 Patient - system

The interactions between the different components of the system for monitoring a patient are depicted in Figure 5. In particular, the figure shows the case where an event produces an alert relevant for the doctor and the patient. For the sake of clarity the figure shows only the interactions when having one sensor and one agent, but a patient can have multiple instances of these components.

In the first place the patient activates the Android application, and turns on the sensors used for the continuous monitoring. At this point the agent environment starts the execution of its life-cycle by waiting for the reception of physiological measurements produced by the sensors. The environment approach of waiting for events rather than being continuously running is intended to extend the battery of the device as much as possible. Once the sensor measures a physiological value, the context entity associated with the sensor encapsulates and forwards it to the environment's queue of events. The environment then activates all the agents interested in that particular event acting thereby as the mediator in the publish/subscribe pattern [15]. The agent body is responsible for perceiving this event from the environment and send it to its agent mind to evaluate if an alert has to be triggered. In the next step the agent mind activates its internal cycle. First, it updates its internal state with the event it just perceived. Second, it tries to achieve its goals by revising the monitoring rules defined for the patient. Finally, in the case that the event triggers a particular rule, the agent creates an action representing the alert to be notified. This action is thrown to the environment through the agent body and notified to the patient through the graphical user interface of the mobile application. The environment also forwards the alert to a context entity that is responsible to redirect it to a web service located at the hospital, so that it can be added to the electronic health records of the patient and notified to the doctor.

2.3.2 Doctor - system

The rules for monitoring a patient are defined by the doctor through a web interface as explained in the corresponding section. In order to use these rules, they must be sent to the agents running in the MAGPIE platform. Figure 6 shows the interactions between the elements involved in this process. In Tier 3 when a new set of monitoring rules are defined by the doctor, these are stored in a local repository and are ready to be downloaded by the monitoring application through a web server. In Tier 2, a Rest client Context Entity is



(a) Single physiological parameter

(b) Repetition of a physiological parameter in a time window

(c) Combination of different physiological parameters in a time window

(d) Meta-rule combining the repetition of a pattern of events in a time window

Fig. 4 Temporal and graphical representations of different monitoring rules

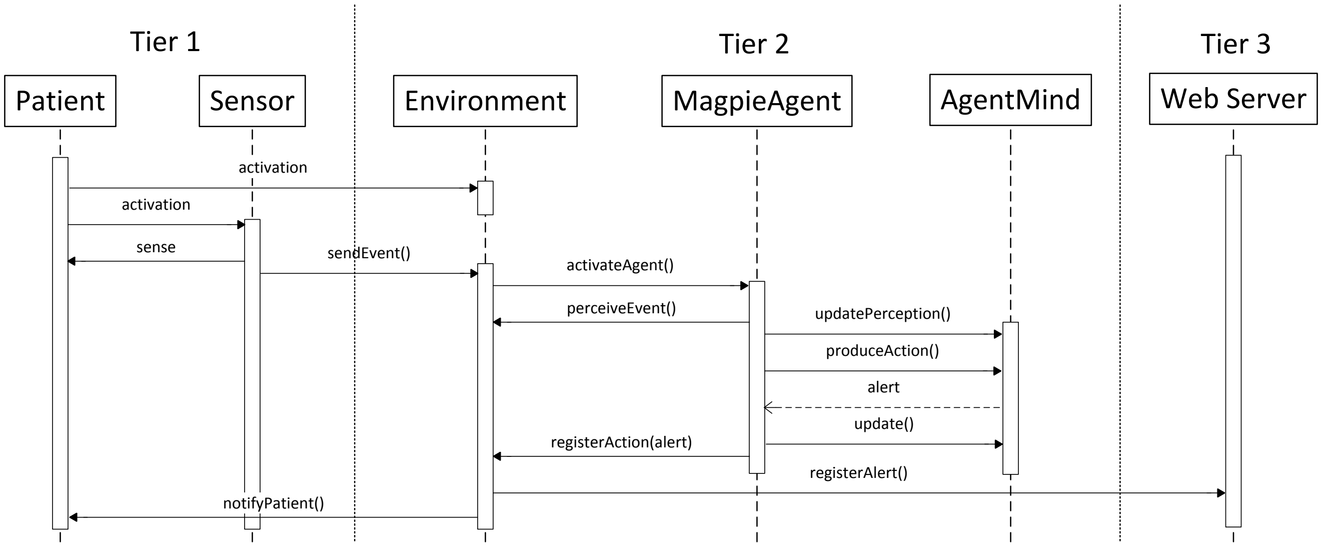


Fig. 5 Interations taken place between the tiers of the system for patient monitoring

responsible for connecting with this web server and getting the monitoring rules. The rules are then encapsulated by the Context Entity as a rule set event and sent to the Agent Environment's queue of events for their processing. Once the Environment is ready for processing events, it activates the agents interested in that particular rule set, which is finally loaded into the agent's mind after the previous rule set has been discarded.

3 Knowledge representation

3.1 Agent mind cycle

Within MAGPIE agents have a mind cycle for perceiving, updating the internal state and then acting in the agent environment following a reactive agent pattern. The specification of the mind cycle is done as follows:

$$\begin{aligned}&\text{agent\_cycle}(T) \leftarrow\\&\qquad \text{perceive}(P,T),\\&\qquad \text{act}(A,T),\\&\qquad \text{update}(A,T),\\&\qquad \text{now}(T_{new}),\\&\qquad \text{agent\_cycle}(T_{new}).\end{aligned} \tag{1}$$

At each cycle, the agent perceives events P coming from the environment, revises its knowledge base, decides for an action A to be performed at time T, and produces the action in the agent environment by pushing it to the agent body. In the next step the agent updates the knowledge base with the knowledge of having performed the action and starts a new cycle at time T_{new}. In more details, the perceive/2 predicate simply asserts events in the agent mind to modify the model that the agent has about the patient. The act/2 predicate simply checks for alerts that hold in the agent mind, and if these hold an action is produced. The update/2 predicate is similar to the perceive/2 predicate, but it rather asserts internal events, such as events linked to actions performed in the agent environment.

3.2 Event calculus

The agent mind cycle uses the Event Calculus (EC) [25] as the underlying formalism to deal with the events produced in the agent environment. The EC is a formalism for representing actions and their effects, and therefore it is suitable to model expert systems representing the evolution in time of an entity by means of the production of events. In MAGPIE, the EC reasoner is embedded inside an agent, and models the monitoring rules applied by the medical doctors through the web interface.

The EC is based on many-sorted first-order predicate calculus, known as domain independent axioms, which are represented as normal logic programs that are executable in Prolog. The underlying time model of the EC is linear. The EC manipulates fluents, where a fluent represents a property which can have different values over time. The term F=V denotes that the fluent F has value V, as a consequence of an action that took place at some earlier time-point and not terminated by another action in the meantime. Table 1 summarizes the main EC predicates used. Predicates, function symbols and constants start with a lower-case letter, while variables starts with an upper-case letter. Predi-

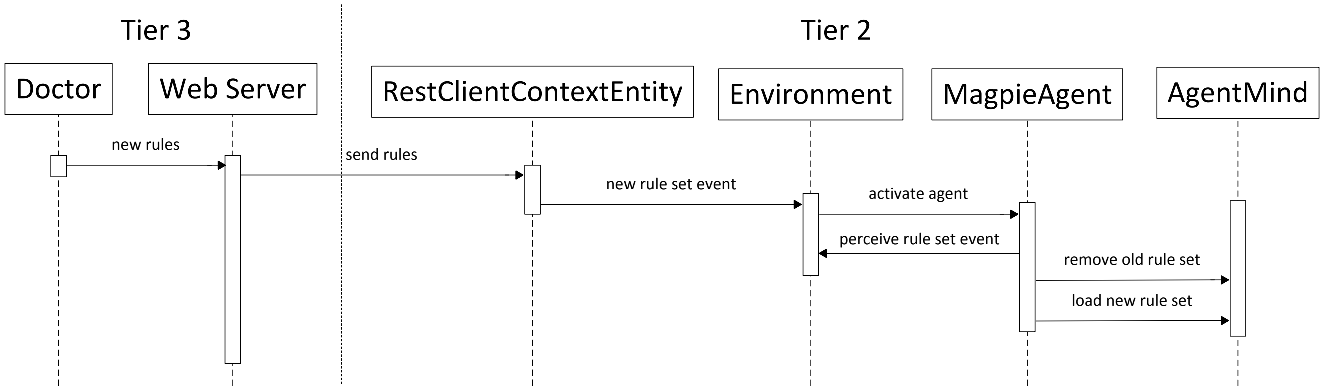


Fig. 6 Interations taken place for updating the monitoring rules

Predicate	**Meaning**
initially(F=V)	The value of fluent F is V at time 0
holdsAt(F=V,T)	The value of fluent F is V at time T
holdsFor(F=V,[Tmin,Tmax])	The value of fluent F is V between Tmin and Tmax
initiatesAt(F=V,T)	At time T the fluent F is initiated to have value V
terminatesAt(F=V,T)	At time T the fluent F is terminated from having the value V
broken(F=V,[Tmin,Tmax])	The value of fluent F is either terminated at Tmax, or initiated to a different value than V between Tmin and Tmax
happensAt(E,T)	An event E takes place at time T updating the state of the fluents

Table 1 Main Event Calculus predicates used

cates are referenced as `predicate/N`, where `predicate` is the name of the predicate and `N` the arity of the predicate, i.e. its number of arguments.

The domain independent axioms of the EC are the following

$$\text{holdsAt}(F = V, 0) \leftarrow \text{initiatially}(F = V) \tag{2}$$

$$\begin{aligned}&\text{holdsAt}(F = V, T) \leftarrow \\ &\qquad \text{initiatesAt}(F = V, T_s), T_s < T, \\ &\qquad \text{not broken}(F = V, [T_s, T])\end{aligned} \tag{3}$$

Predicate (2) states that a fluent F holds value V at time 0, if it has been initially set to this value. For any other time T > 0, the predicate (3) states that the fluent holds at time T if it has been initiated to value V at some earlier time point, and it has not been broken on the meanwhile.

$$\begin{aligned}&\text{broken}(F = V, [Tmin, Tmax]) \leftarrow \\ &\qquad \text{terminatesAt}(F = V, T), Tmin < T, Tmax > T\end{aligned} \tag{4}$$

$$\begin{aligned}&\text{broken}(F = V_1, [Tmin, Tmax]) \leftarrow \\ &\qquad \text{initiatesAt}(F = V_2, T_i), V_1 \neq V_2, \\ &\qquad Tmin < Ti, Tmax > Ti\end{aligned} \tag{5}$$

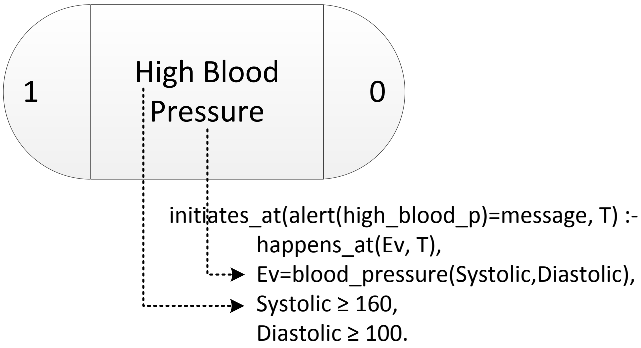
Predicates (4) and (5) specify the conditions that brake a fluent. Predicate (4) states that a fluent is broken between two time points Tmin and Tmax if within this interval it has been terminated to have value V. Alternatively, predicate (5) states that a fluent is broken within a time interval if it has been initiated to hold a different value.

$$\begin{aligned}&\text{holdsFor}(F = V, [Tmin, Tmax]) \leftarrow \\ &\qquad \text{initiatesAt}(F = V, Tmin), \\ &\qquad \text{terminiatesAt}(F = V, Tmax), \\ &\qquad \text{not broken}(F = V, [Tmin, Tmax])\end{aligned} \tag{6}$$

$$\begin{aligned}&\text{holdsFor}(F = V, [Tmin, infPlus]) \leftarrow \\ &\qquad \text{initiatesAt}(F = V, Tmin), \\ &\qquad \text{not broken}(F = V, [Tmin, infPlus])\end{aligned} \tag{7}$$

$$\begin{aligned}&\text{holdsFor}(F = V, [infMin, Tmax]) \leftarrow \\ &\qquad \text{terminatesAt}(F = V, Tmax), \\ &\qquad \text{not broken}(F = V, [infMin, Tmax])\end{aligned} \tag{8}$$

Predicates (6), (7) and (8) deal with the validity intervals of fluents. In particular, predicate (6) specifies that a fluent F keeps value V for a time interval going from Tmin to Tmax if nothing happens in the middle that breaks such an interval. Predicates (7) and (8) behave in the same way, but deal with open intervals.



```
initiates_at(alert(high_blood_p)=message, T) :-
    happens_at(Ev, T),
    Ev=blood_pressure(Systolic,Diastolic),
    Systolic ≥ 160,
    Diastolic ≥ 100.
```

Fig. 7 Relations between the graphical representation of an event and Prolog code in Event Calculus

The domain dependent predicates in EC are typically expressed in terms of the `initiatesAt/2` and `terminatesAt/2` predicates. One example of a common rule for `initiatesAt/2` is

$$\begin{aligned} &\text{initatesAt}(F = V, T) \leftarrow \\ &\qquad \text{happensAt}(Ev, T), \\ &\qquad Conditions[T] \end{aligned} \tag{9}$$

The above definition states that a fluent is initiated to value V at time T if an event Ev happens at this time point, and some optional conditions depending on the domain are satisfied. In MAGPIE, these events that must happen are physiological measurements from the patient.

3.3 Relation between graphical rules and EC rules

To allow agents to produce alerts according to the rules defined by the doctors, the graphical rules are converted to Prolog predicates that are expressed in terms of EC clauses. Figure 7 shows how the elements defined by an event are linked to its Prolog representation. The monitoring rules are modeled using the `initiates_at/2` EC predicate, which means that an alert of a particular type is triggered at time T if the events defined in the body rule happens. In the particular example in the Figure 7, an alert of type *high_blood_p* notifies a *message*, if at time T there is a blood pressure measurement whose systolic value is higher or equal to 160 mmHg and its diastolic value is higher or equal to 100 mmHg.

In the case of having a monitoring rule with more than one event, it can be derived from the previous example that the different events can be nested taking into account the time in which they happen. Thus, a sequential rule of three events defining the glucose pattern: *high* → *low* → *high*, in a time period of one day is represented in Prolog as follows

$$\begin{aligned} &\text{initiatesAt}(\text{alert}(\text{glucose\_pattern}) = \text{message}, T) \leftarrow \\ &\qquad \text{happensAt}(Ev1, Tev1), \\ &\qquad \text{happensAt}(Ev2, Tev2), \\ &\qquad \text{happensAt}(Ev3, Tev3), \\ &\qquad Ev1 = \text{glucose}(Value1), \\ &\qquad Ev2 = \text{glucose}(Value2), \\ &\qquad Ev3 = \text{glucose}(Value3), \\ &\qquad Value1 \geq 8, \\ &\qquad Value2 \leq 3.8, \\ &\qquad Value3 \geq 8, \\ &\qquad Tev3 > Tev2, \\ &\qquad Tev2 > Tev1, \\ &\qquad \text{last\_day}(Tev1, Tev3). \end{aligned} \tag{10}$$

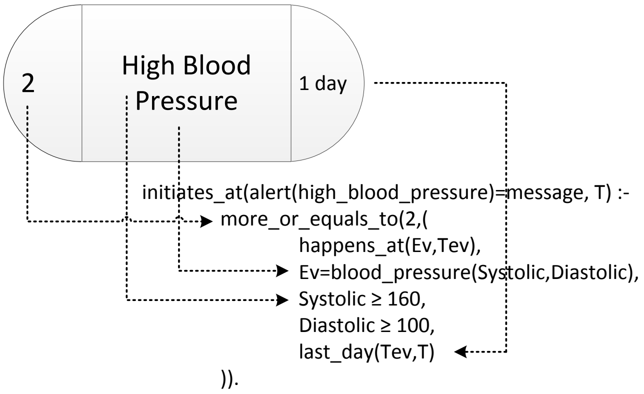
Where the predicate `last_day/2` computes if the distance in time between the two arguments is less than one day. Similar Prolog predicates exist for the different time windows that can be specified in the web interface.

$$\begin{aligned} &\text{last\_day}(Tev, Tfinal) \leftarrow \\ &\qquad Tinit \text{ is } Tfinal - 24 * 3600 * 1000, \\ &\qquad Tev \leq Tfinal, \\ &\qquad Tev \geq Tinit. \end{aligned} \tag{11}$$

However, this approach is not practical for defining complex rules as a graphical rule composed with many events will derive in a set of different EC predicates each one corresponding to a temporal permutation of all the events. To deal with this issue a `more_or_equals_to/2` predicate has been defined. This predicate counts the number of facts in the knowledge base satisfying the conditions defined in the second argument, and returns true if it finds at least the same number of facts defined in the first argument. As shown in Figure 8 this predicate is used in the body of the `initiates_at/2` predicate to count the number of events satisfying the defined conditions. In the example, an alert is triggered if in the last day there were at least two blood pressure events, whose systolic and diastolic values were higher or equal to 160 mmHg and 100 mmHg respectively. The definition of the predicate `more_or_equals_to/2` is as follows

$$\begin{aligned} &\text{more\_or\_equals\_to}(Number, Expr) \leftarrow \\ &\qquad \text{findall}(\_, Expr, List), \\ &\qquad \text{length}(List, Val), \\ &\qquad Val \geq Number. \end{aligned} \tag{12}$$

In the case of meta-rules, where the goal is to alert the repetition of a particular pattern of events in a given period of time, two different domain dependent EC predicates are created: the inner part and the outer part. The inner part, shown in Figure 9(a) represents a



```
initiates_at(alert(high_blood_pressure)=message, T) :-
        more_or_equals_to(2,(
                happens_at(Ev,Tev),
                Ev=blood_pressure(Systolic,Diastolic),
                Systolic ≥ 160,
                Diastolic ≥ 100,
                last_day(Tev,T)
        )).
```

Fig. 8 Use of the `more_or_equals_to/2` counting predicate in a complex rule

single occurrence of the pattern defined, and the predicate consists on the nesting of `more_or_equals_to/2` predicates. The outer part (Figure 9(b)) consists on counting if the specified pattern of events happens at least a certain number of times in the given time window. The counting is done with the `more_or_equals_to/2` predicate, where the event is an `alert_sent` representing the inner part of the rule.

3.4 Specific DM rules

Observing blood glucose trends and patterns in diabetic patients has been reported to be beneficial [41], as it can help to address the cause of the problem. In here the following patterns of interest have been selected

- Pattern 1: Brittle diabetes, defined as a glucose rebound going from less than 3.8 mmol/l to more than 8.0 mmol/l in a period of six hours

$$\begin{aligned}
&\text{initiatesAt}(alert(p1) = \text{'brittle diabetes'}, T) \leftarrow\\
&\qquad \text{happensAt}(Ev1, Tev1),\\
&\qquad \text{happensAt}(Ev2, Tev2),\\
&\qquad Ev1 = \text{glucose}(Value1),\\
&\qquad Ev2 = \text{glucose}(Value2),\\
&\qquad Value1 \leq 3.8,\\
&\qquad Value2 \geq 8,\\
&\qquad Tev2 > Tev1,\\
&\qquad \text{last\_six\_hours}(Tev1, Tev2).
\end{aligned} \tag{13}$$

- Pattern 2: Pre-hypertension, defined as two events of high blood pressure in a period of one week.

$$\begin{aligned}
&\text{initiatesAt}(alert(p2) = \text{'pre-hypertension'}, T) \leftarrow\\
&\qquad \text{more\_or\_equals\_to}(2, (\\
&\qquad \text{happens\_at}(Ev1, Tev),\\
&\qquad Ev1 = \text{blood\_pressure}(Sys, Dias),\\
&\qquad Sys \geq 130,\\
&\qquad Dias \geq 80,\\
&\qquad \text{last\_week}(Tev, T))).
\end{aligned} \tag{14}$$

- Pattern 3: Gaining weight, defined as going from a 2% weight lost to a 1% gain weight. This pattern is dynamic for each patient, and for an initial weight of 111.5 kg the thresholds are as follows.

$$\begin{aligned}
&\text{initiatesAt}(alert(p3) = \text{'gaining weight'}, T) \leftarrow\\
&\qquad \text{happensAt}(Ev1, Tev1),\\
&\qquad \text{happensAt}(Ev2, Tev2),\\
&\qquad Ev1 = \text{weight}(Value1),\\
&\qquad Ev2 = \text{weight}(Value2),\\
&\qquad Value1 \geq 109.3,\\
&\qquad Value2 >= 110.4,\\
&\qquad Tev2 > Tev1,\\
&\qquad \text{last\_week}(Tev1, Tev2).
\end{aligned} \tag{15}$$

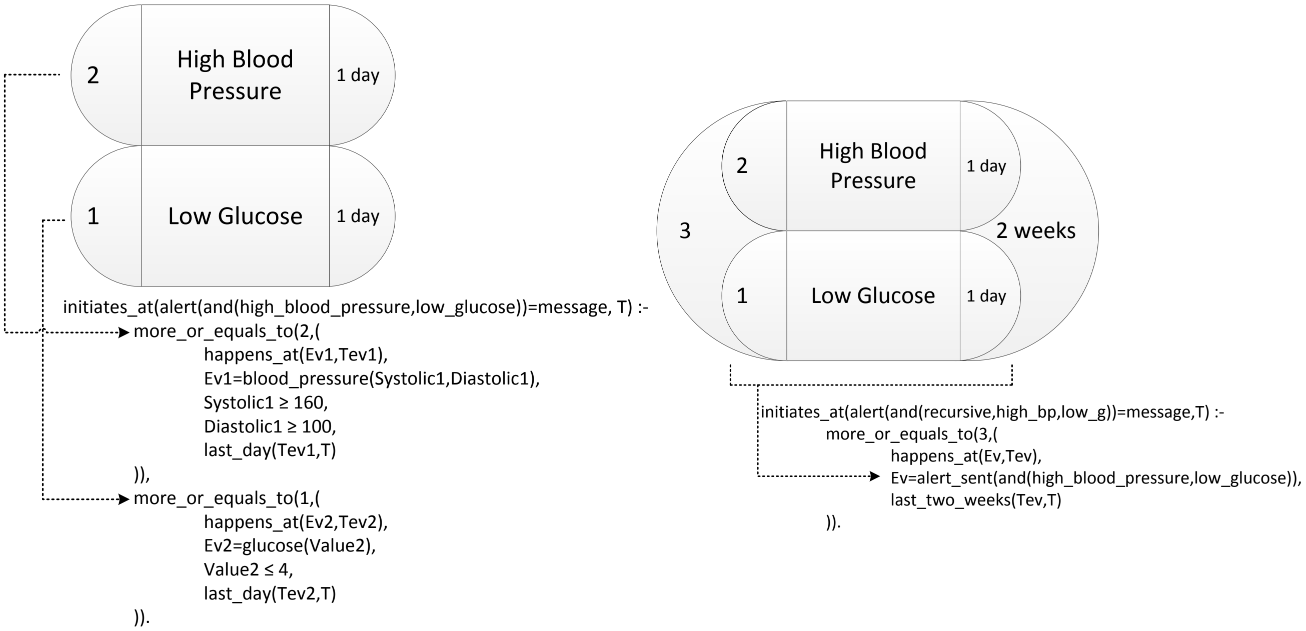
4 Evaluation

The evaluation of MAGPIE has been done in two different ways. First, the ability of the system to detect composite events on patients with diabetes type II has been measured according to the rules specified in the previous section. Second, a series of simulations have been conducted to determine how scalable is the MAGPIE approach in comparison with a centralized approach.

4.1 Recognition of event patterns

A retrospective analysis on COMMODITY_{12} data has been done for evaluating the ability of the system in detecting patterns in real data. Table 2 summarizes the statistics of the dataset. The dataset consists on 21 diabetic patients that used the sensors described in Section 2 during a period of six weeks approximately. The rows in samples per day indicate how well the patients follow the treatment. Ideally these values should be 1.71±1.89, 1±0, 1±0 for the glucose, blood pressure and weight respectively.

Table 3 shows the results concerning the triggering of the rules given the three previously presented selected patterns. The interesting aspect of these results is that given the patients data, the system is able to find the patterns defined by the doctors. In particular the selected patterns where rather simple and based on the



```
initiates_at(alert(and(high_blood_pressure,low_glucose))=message, T) :-
          more_or_equals_to(2,(
                    happens_at(Ev1,Tev1),
                    Ev1=blood_pressure(Systolic1,Diastolic1),
                    Systolic1 ≥ 160,
                    Diastolic1 ≥ 100,
                    last_day(Tev1,T)
          )),
          more_or_equals_to(1,(
                    happens_at(Ev2,Tev2),
                    Ev2=glucose(Value2),
                    Value2 ≤ 4,
                    last_day(Tev2,T)
          )).
```

(a) Nesting of events in a complex rule

```
initiates_at(alert(and(recursive,high_bp,low_g))=message,T) :-
          more_or_equals_to(3,(
                    happens_at(Ev,Tev),
                    Ev=alert_sent(and(high_blood_pressure,low_glucose)),
                    last_two_weeks(Tev,T)
          )).
```

(b) Meta-rule composed with the complex rule

Fig. 9 Examples of code produced for a complex rule and a meta-rule

Patient	Days	Samples per Day (Mean±SD)			Measurement (Mean±SD)			
		Glucose	Blood Pressure	Weight	Glucose (mmol/l)	Sys. BP (mmHg)	Dias. BP (mmHg)	Weight (kg)
205	42	1.71±1.96	2.33±1.14	0.95±0.21	7.45±1.64	128.06±8.27	77.04±4.93	108.71±1.43
209	44	1.69±1.88	1.86±0.42	0.98±0.15	10.53±1.72	143.78±9.68	79.63±6.50	84.77±0.98
213	42	1.95±2.23	1.93±0.41	1.00±0.31	8.24±1.76	126.06±7.41	73.42±4.86	94.95±0.81
217	42	1.57±1.81	1.90±0.37	0.97±0.15	9.31±2.27	128.49±7.59	66.98±6.40	62.67±0.39
221	42	2.00±2.26	2.36±0.66	1.07±0.26	6.85±1.64	150.63±9.28	89.04±8.70	106.34±0.57
225	41	1.66±1.87	1.83±0.44	1.00±0.00	9.49±2.40	143.52±12.48	80.95±7.92	93.70±0.43
229	44	1.93±2.15	3.39±2.17	1.05±0.37	7.31±1.69	126.52±8.10	81.10±5.42	78.50±0.85
233	34	1.65±1.69	1.82±0.39	1.03±0.17	7.06±1.34	133.56±8.34	69.68±7.51	83.51±0.52
237	31	1.84±1.95	1.97±0.55	0.97±0.18	5.68±1.06	139.87±12.91	79.70±5.98	85.47±0.56
241	14	1.86±2.03	1.93±0.27	1.00±0.00	9.86±2.73	126.93±7.19	68.96±3.39	89.63±0.48
261	32	1.88±1.70	1.91±0.39	1.16±0.77	6.21±1.28	134.90±12.67	81.05±9.10	67.73±0.51
265	42	1.60±1.60	1.98±0.41	0.98±0.15	9.27±1.99	120.54±7.78	68.43±5.69	86.80±0.39
269	42	1.71±1.84	1.90±0.48	0.90±0.30	10.21±3.36	136.83±10.44	81.16±6.82	77.51±0.61
273	30	1.73±1.64	1.97±0.56	0.97±0.41	6.92±1.02	121.20±6.63	67.85±4.77	99.16±0.43
277	42	1.69±1.73	1.83±0.58	0.98±0.15	6.39±1.40	90.77±6.85	61.29±5.69	58.12±0.68
281	42	1.67±1.80	1.98±0.41	1.05±0.22	7.52±1.49	127.92±6.90	79.08±5.45	98.39±1.22
285	40	1.55±1.71	1.70±0.52	1.00±0.45	11.35±2.41	127.49±7.50	82.07±5.16	84.47±0.81
289	36	1.92±1.80	2.00±0.93	1.00±0.00	10.32±2.95	156.28±11.68	92.25±7.62	116.61±0.83
293	41	1.73±1.83	2.00±0.39	1.00±0.00	6.05±0.96	124.74±8.25	70.41±5.12	95.64±1.21
297	38	1.84±1.90	1.76±0.63	1.00±0.00	6.67±1.12	120.34±8.12	54.61±7.16	68.59±1.05
529	32	1.34±1.12	1.97±0.59	1.13±0.61	9.10±2.48	137.81±9.21	79.83±5.26	95.33±0.46

Table 2 Statistics describing the dataset

Patient	Pattern 1	Pattern 2	Pattern 3
205	0	19	0
209	0	44	0
213	0	4	0
217	0	0	6
221	0	86	5
225	0	38	0
229	0	41	0
233	0	7	0
237	0	27	3
241	2	0	0
261	0	35	4
265	0	0	0
269	0	39	0
273	0	0	0
277	0	0	9
281	0	19	17
285	0	12	0
289	0	67	11
293	0	0	0
297	0	0	20
529	0	25	0

Table 3 Detection of patterns in data

common practice of the medical doctors, so the system could detect the entirety of the selected patterns. This is significant because the ability to detect these patterns allow medical doctors to modify the treatment of a selected patient and thus the reaction time of the medical doctors is more effective.

4.2 Scalability test

The typical approach to analyze the patient's monitoring data in PHSs is centralized, as it is all analyzed in the Tier 3. In contrast, with MAGPIE the computation to trigger the alerts is done in the Tier 2, so that it is distributed among the patients' smartphones. Thus, the Tier 3 is free from doing this task, although the alerts must be sent to Tier 3 to notify them to the doctor.

To measure the scalability of both approaches, a series of simulations have been realized using the Amazon Web Services [2]. To compare the performance of the two approaches, eleven Amazon EC2 instances of the same type t2.micro have been used. Ten instances for running the clients representing the patient's base station in Tier 2 and one for running the monitoring server in Tier 3. Thus, in terms of hardware the clients and the server are always identical for both approaches.

The simulations consists on distributing a certain number of clients over the ten EC2 instances, which run simultaneously performing the same operations. For the simulation of the MAGPIE approach, each client runs its own agent, which it is forced to trigger an alert composed by two events and send it to the remote monitoring server. The server stores the alert in a repository and sends back an acknowledge message to the client. Whereas in the centralized approach, despite each client having its own agent, all the agents run in the server. In this approach, each client sends two events that force its agent to trigger an alert that is first stored in a repository and then notified back to the client as an acknowledgement message. In both cases the latency is measured as the elapsed time from the generation of the first event until the client is notified with the acknowledgment.

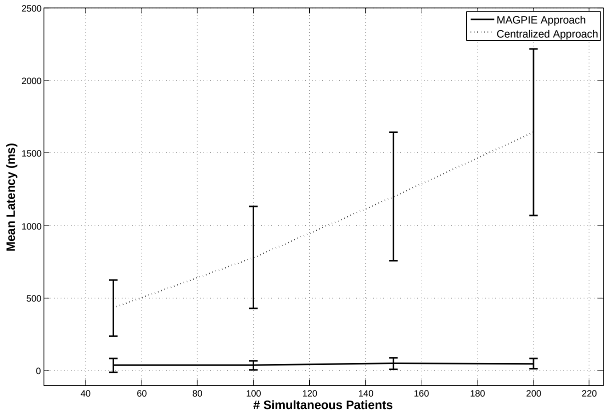


Fig. 10 Latency when an alert composed by two events is triggered

Figure 10 shows the results of the simulation, which suggest that computation capabilities of smartphones should be used in PHSs. While the MAGPIE approach has a flat response at around 50 ms. from 50 to 200 simultaneous patients, the centralized approach increases linearly with the number of patients. Moreover, when the number of simultaneous patients is 250 patients some clients experiences a timeout.

5 Related work

The work of this paper applies agent technology in the domain of healthcare to monitor chronic patients suffering from DM. Other heathcare systems also use agents to perform different tasks like coordinating data exchange between healthcare institutions [29,39], or to support people in Ambient Assisted Living [6,27]. Other systems and uses of agents in healthcare are reviewed in [21].

In the context of PHSs, MADIP [36] is a relevant work that is build on top of the JADE agent platform. In this system, different kind of agents are used with the

[2] http://aws.amazon.com/

main purpose of monitoring vital signs and to inform healthcare professionals when abnormal situations are detected. However, the use of software for desktop computers can limit the mobility of the patient. A similar monitoring system is presented in [3] where data from sensors is analyzed locally in a PDA but in the context of arrhythmia detection. The monitoring rules used in this system are inferred form classifiers and hard coded in the PDA. Contrary, in MAGPIE rules are based on the expertise of the medical doctor who can define and change them at runtime. Rules for monitoring patients are also used in [33], where a conceptual model of multi-agent patient care management is described. The model splits the medical knowledge between different agents, however it is based on IF-THEN rules that do not have the expressiveness of a logic programming approach.

In relation with DM monitoring some proposals complement the application described in this paper by targeting different goals. For example to help patients with type 2 diabetes on achieving and maintaining healthy lifestyles by estimating their daily calories balance [17]. A smartphone based system to asses diabetic patients suffering from foot ulcers [42]; or an application to monitor blood glucose levels and insulin injection, which also suggests exercise goals [18].

A different approach to manage DM is referred as the artificial pancreas [13]. This approach combines three different elements to emulate the functionality of a real pancreas: a continuous glucose monitoring (CGM) device, an insulin pump and an intelligent algorithm. Within this context several algorithms have been proposed to predict blood glucose levels in a given time horizon, in order to calculate the right dose of insulin to be injected by the insulin pump. These prediction algorithms are based on different techniques like rules [12], artificial neural networks [32], proportional integral derivative controllers [28], and neuro-fuzzy techniques [45]. There are also algorithms focusing on denoising CGM readings [16] that can help on improving the prediction. The artificial pancreas as a method to manage DM has the drawback that insulin pumps are invasive devices which can cause skin infection and dermatological changes at the site of infusion [37], and problems can occur with blocked, kinked or leaking cannulas [19]. Moreover, pumps do not send electronically to the doctor injected insulin doses, so that the possibility to adapt the treatment is limited by the times that the doctor can visit the patient. Another issue with respect CGM devices is that they provide glucose values measured in the interstitial fluid so there is a delay of about ten minutes between the measurement and the plasma glucose [23].

6 Conclusions

In this paper we presented an agent based distributed PHS to monitor patients affected by chronic illnesses. We evaluated the system with respect to its scalability by comparing it with a centralised approach and we evaluated its capability of detecting common pathological patterns in patients affected by diabetes type 2. The conclusion is that the best practice towards such systems is to move the reasoning on the mobile part of the PHS, and to allow to modify the rules for the detection of the pathological patterns at runtime. As future work we plan to:

- model more infrequent pathological patterns;
- introduce prediction capabilities with predictive rules;
- create an interface for temporal rule learning.

Finally, from the medical side, we also plan to apply this system on different physiological values and different illnesses than diabetes type 2.

Acknowledgements This work was partially supported by Hasler Foundation, HES-SO RCSO ISNet and the FP7 287841 COMMODITY12 project.

References

1. Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., Vecchio, A.: A smartphone-based fall detection system. Pervasive Mob. Comput. **8**(6), 883 – 899 (2012)
2. Arnrich, B., Mayora, O., Bardram, J., Tröster, G.: Pervasive healthcare. Paving the way for a pervasive, user-centered and preventive healthcare model. Methods Inf. Med. **49**(1), 67–73 (2010)
3. Bagüés, M.I., Bermúdez, J., Burgos, A., Goñi, A., Illarramendi, A., Rodríguez, J., Tablado, A.: An innovative system that runs on a PDA for a continuous monitoring of people. In: Proc. 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS), pp. 151–156 (2006)
4. Bardram, J.E.: Pervasive healthcare as a scientific discipline. Methods Inf. Med. **47**(3), 178–185 (2008)
5. Baumgarten, M., Mulvenna, M.: Cognitive sensor networks: Towards self-adapting ambient intelligence for pervasive healthcare. In: Proc. 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 366–369 (2011)
6. Borowczyk, A., Gawinecki, M., Paprzycki, M.: BDI agents in a patient monitoring scenario. In: Proc. 2nd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 82–85 (2008)
7. Bromuri, S., Stathis, K.: Distributed agent environments in the Ambient Event Calculus. In: Proc. 3rd ACM International Conference on Distributed Event-Based Systems (DEBS) (2009)
8. Brugués, A., Bromuri, S., Pegueroles, J., Schumacher, M.: MAGPIE: An agent platform for the development of mobile applications for pervasive healthcare. In: Proc. 3rd

International Workshop on Artificial Intelligence and Assistive Medicine (AI-AM/NetMed), pp. 6–10 (2014)

9. Brugués, A., Bromuri, S., Pegueroles, J., Schumacher, M.: Providing interoperability to a pervasive healthcare system through the HL7 CDA standard. In: Proc. 15th International HL7 Interoperability Conference (IHIC), p. to appear (2015)
10. Buse, J.B., Ginsberg, H.N., Bakris, G.L., Clark, N.G., Costa, F., Eckel, R., Fonseca, V., Gerstein, H.C., Grundy, S., Nesto, R.W., et al.: Primary prevention of cardiovascular diseases in people with diabetes mellitus a scientific statement from the American Heart Association and the American Diabetes Association. Diabetes care **30**(1), 162–172 (2007)
11. Cacciagrano, D., Corradini, F., Culmone, R., Merelli, E., Vito, L.: Healthcare tomorrow: Toward self-adaptive, ubiquitous and personalized services. In: Proc. 5th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), pp. 245–250 (2011)
12. Capel, I., Rigla, M., García-Sáez, G., Rodríguez-Herrero, A., Pons, B., Subías, D., García-García, F., Gallach, M., Aguilar, M., Pérez-Gandía, C., et al.: Artificial pancreas using a personalized rule-based controller achieves overnight normoglycemia in patients with type 1 diabetes. Diabetes Technol. Ther. **16**(3), 172–179 (2014)
13. Cobelli, C., Renard, E., Kovatchev, B.: Artificial pancreas: past, present, future. Diabetes **60**(11), 2672–2682 (2011)
14. Dolin, R.H., Alschuler, L., Boyer, S., Beebe, C., Behlen, F.M., Biron P. V .and Shabo, A.: HL7 Clinical document drchitecture, release 2. J. Am. Med. Inf. Assoc. **13**(1), 30–39 (2006)
15. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
16. Facchinetti, A., Sparacino, G., Cobelli, C.: An online self-tunable method to denoise cgm sensor data. IEEE Trans. Biomed. Eng. **57**(3), 634–641 (2010)
17. Fico, G., Fioravanti, A., Arredondo, M., Ardigo, D., Guillen, A.: A healthy lifestyle coaching-persuasive application for patients with type 2 diabetes. In: Proc. 32nd International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2221–2224 (2010)
18. Gislason, B., Mcknight, C., Potvin, B., Stuart, S., Zepeda, J., Weber, J., Elmiligi, H.: Introducing glucofit: An assistive technology for monitoring and managing diabetes. In: Proc. 7th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA), pp. 414–419 (2012)
19. Heinemann, L., Krinelke, L.: Insulin infusion set: the achilles heel of continuous subcutaneous insulin infusion. J Diabetes Sci Technol **6**(4), 954–964 (2012)
20. Institute of Electrical and Electronics Engineers: IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE Std 610 pp. 1–217 (1991)
21. Isern, D., Sánchez, D., Moreno, A.: Agents applied in health care: A review. Int. J. Med. Inform. **79**(3), 145 – 166 (2010)
22. Kafalı, Ö., Bromuri, S., Sindlar, M., van der Weide, T., Aguilar Pelaez, E., Schaechtle, U., Alves, B., Zufferey, D., Rodriguez-Villegas, E., Schumacher, M.I., Stathis, K.: COMMODITY$_{12}$: A smart e-health environment for diabetes management. J. Ambient Intell. Smart Environ. **5**(5), 479–502 (2013)
23. Keenan, D.B., Mastrototaro, J.J., Voskanyan, G., Steil, G.M.: Delays in minimally invasive continuous glucose monitoring devices: a review of current technology. J Diabetes Sci Technol **3**(5), 1207–1214 (2009)
24. Kim, S.H., Jeon, J.W.: Programming LEGO mindstorms NXT with visual programming. In: Proc. 2007 International Conference on Control, Automation and Systems (ICCAS), pp. 2468–2472 (2007)
25. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Gener. Comput. **4**(1), 67–95 (1986)
26. Mukherjee, A., Pal, A., Misra, P.: Data analytics in ubiquitous sensor-based health information systems. In: Proc. 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST), pp. 193–198 (2012)
27. Nefti, S., Manzoor, U., Manzoor, S.: Cognitive agent based intelligent warning system to monitor patients suffering from dementia using ambient assisted living. In: Proc. 2010 International Conference on Information Society (i-Society), pp. 92–97 (2010)
28. O'Grady, M.J., Retterath, A.J., Keenan, D.B., Kurtz, N., Cantwell, M., Spital, G., Kremliovsky, M.N., Roy, A., Davis, E.A., Jones, T.W., et al.: The use of an automated, portable glucose control system for overnight glucose control in adolescents and young adults with type 1 diabetes. Diabetes care **35**(11), 2182–2187 (2012)
29. Palazzo, L., Rossi, M., Dragoni, A.F., Claudi, A., Dolcini, G., Sernani, P.: A multi-agent architecture for health information systems. In: Proc. 7th KES Conference on Agent and Multi-Agent Systems: Technologies and Applications (KES-AMSTA), pp. 375–384 (2013)
30. Parsons, S., Sklar, E.: Teaching AI using LEGO mindstorms. In: Proc. AAAI Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education, pp. 8–13 (2004)
31. Peng, H., Hu, B., Liu, Q., Dong, Q., Zhao, Q., Moore, P.: User-centered depression prevention: An eeg approach to pervasive healthcare. In: Proc. 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 325–330 (2011)
32. Pérez-Gandía, C., Facchinetti, A., Sparacino, G., Cobelli, C., Gómez, E.J., Rigla, M., De Leiva, A., Hernando, M.E.: Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring. Diabetes Technol. Ther. **12**(1), 81–88 (2010)
33. Shankararaman, V., Ambrosiadou, V., Loomes, M., Panchal, T.: Patient care management using a multi-agent approach. In: Proc. 2000 International Conference on Systems, Man, and Cybernetics (SMC), vol. 3, pp. 1817–1821 (2000)
34. Shaw, J., Sicree, R., Zimmet, P.: Global estimates of the prevalence of diabetes for 2010 and 2030. Diabetes Res. Clin. Pract. **87**(1), 4–14 (2010)
35. Stathis, K., Kakas, A., Lu, W., Demetriou, N., Endriss, U., Bracciali, A.: PROSOCS: a platform for programming software agents in computational logic. In: Proc. 4th International Symposium "From Agent Theory to Agent Implementation" (AT2AI-4), pp. 523–528 (2004)
36. Su, C.J., Wu, C.Y.: JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. Appl. Soft. Comput. **11**(1), 315–325 (2011)
37. Tauschmann, M., Hovorka, R.: Insulin pump therapy in youth with type 1 diabetes: toward closed-loop systems. Expert Opin. Drug Deliv. **11**(6), 943–955 (2014)
38. Touati, F., Tabish, R.: U-healthcare system: State-of-the-art review and challenges. J. Med. Syst. **37**(3), 9949 (2013)

39. Urovi, V., Olivieri, A.C., Brugués, A., Bromuri, S., Fornara, N., Schumacher, M.I.: Secure P2P cross-community health record exchange in IHE compatible systems. Int. J. Artif. Intell. Tools **23**(1), 1440,006 (2014)
40. Varshney, U.: Pervasive healthcare and wireless health monitoring. Mobile Netw. Appl. **12**(2-3), 113–127 (2007)
41. Wallymahmed, M.: Encouraging people with diabetes to get the most from blood glucose monitoring: Observing and acting upon blood glucose patterns. Journal of Diabetes Nursing **17**(1), 6–13 (2013)
42. Wang, L., Pedersen, P., Strong, D., Tulu, B., Agu, E., Ignotz, R.: Smartphone based wound assessment system for patients with diabetes. IEEE Trans. Biomed. Eng. (in press)
43. Weyns, D., Omicini, A., Odell, J.: Environment as a first class abstraction in multiagent systems. Auton. Agents Multi-Agent Syst. **14**(1), 5–30 (2007)
44. Wooldridge, M.: An introduction to multiagent systems. John Wiley & Sons (2009)
45. Zarkogianni, K., Mitsis, K., Arredondo, M.T., Fico, G., Fioravanti, A., Nikita, K.: Neuro-fuzzy based glucose prediction model for patients with type 1 diabetes mellitus. In: Proc. 2nd IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 252–255 (2014)
46. Zhang, P., Zhang, X., Brown, J., Vistisen, D., Sicree, R., Shaw, J., Nichols, G.: Global healthcare expenditure on diabetes for 2010 and 2030. Diabetes Res. Clin. Pract. **87**(3), 293 – 301 (2010)