Demonstration of MAGPIE: An Agent Platform for Monitoring Chronic Diseases on Android

Albert Brugués[1], Stefano Bromuri[2], and Michael Schumacher[1]

[1] Applied Intelligent Systems Laboratory,
University of Applied Sciences Western Switzerland, Sierre, Switzerland
`{name.surname}@hevs.ch`
[2] Management Science and Technology,
Open University of the Netherlands, Heerlen, The Netherlands
`stefano.bromuri@ou.nl`

Abstract. In this demonstration, we show how the MAGPIE agent platform works. The aim of this platform is to help on the development of Personal Health Systems (PHSs) for monitoring chronic diseases. The agents of the platform use a symbolic reasoning approach to formalize the events happening to the patient. We developed an Android application based on MAGPIE where we formalized the reasoning for monitoring patients affected by diabetes mellitus.

1 Introduction

Personal Health Systems (PHSs) consist on the decentralization of healthcare services by approaching sampling technologies into the hands of the patients, with the aim of involving them in the management of their illnesses and in their own well being. This way of providing healthcare services removes time and physical barriers and enables the paradigm of *healthcare to anyone, anytime and anywhere* [4]. The use of PHSs has been reported as a prominent way to face the healthcare expenditures due to the increase of life expectancy and its assosiated prevalence of chronic diseases [3].

As shown in Fig. 1, the typical architecture of a PHS consists on three tiers, namely: Tier 1 Body Area Network (BAN), Tier 2 Personal Server and Tier 3 Remote Server. The BAN consists in a set of sensors deployed in the body to collect physiological parameters of the patient, which are transmitted to the Personal Server. The Personal Server is usually a mobile device (smartphone or tablet) with network connectivity that aggregates the data and transmits them to the Remote Server. This last component provides assistance to patients and medical doctors for the management of the disease.

The application of agent technology in PHSs simplifies the modeling of medical knowledge, as agents are autonomous software entities that pursue a set of goals in an intelligent way by applying artificial intelligence 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 patients' physiological parameters.

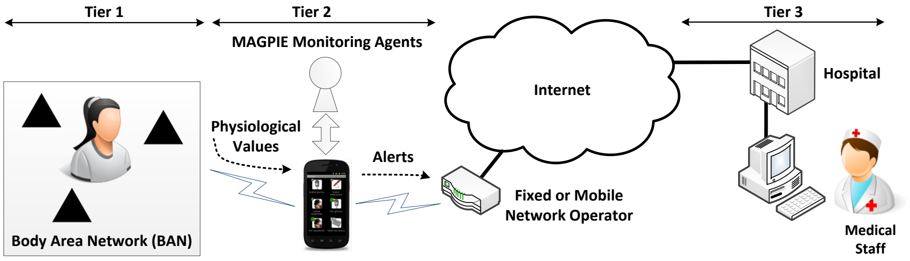

Fig. 1. Architecture of a PHS developed with the MAGPIE agent platform

2 Main Purpose

In the context of PHS, we developed the MAGPIE agent platform [1] as an Android framework for the development of the Tier 2 of PHSs. MAGPIE links the concept of an agent environment in multi-agent systems (MAS), with the patient's environment in PHSs in the sense that monitoring sensors become a source of health information that can be exploited by agents to track the health status of the patient, and perform an action when a potentially dangerous situation is detected. The source code of the platform is available in GitHub[1].

MAGPIE models three different components that can be mapped to the elements of a publish/subscribe system. These are: i) *agents*, as subscriber entities to events happening into the environment, which are responsible to monitor the health status of the patient; ii) *context entities*, as abstractions that encapsulate a source of information from the real world, such as a sensor, and publish into the environment patient-related events like physiological measurements; and the iii) *environment*, which is an entity that act as an event service by mediating the interactions taking place between context entities and agents. This design strategy, based on the publish/subscribe pattern, shields agents and context entities from knowing the implementation details about each other.

More in details, agents are composed by two main parts: a body and a mind. The agent body situates the agent mind in the environment; while the agent mind is the cognitive part of the agent, and is responsible to produce actions according to the perceived events and how the medical knowledge is modeled in the mind. Such medical knowledge is based on temporal reasoning. In particular, the mind models temporal patterns that combine different types of events by exploiting the properties of Event Calculus (EC) [2]. EC is a logic formalism for representing actions and their effects in time. Therefore, it is suitable for modeling expert systems representing the evolution in time of an entity by means of the production of events. EC is based on many-sorted first-order logic predicate calculus, known as domain independent axioms, which can be represented as normal logic programs executable in Prolog.

---

[1] `https://github.com/aislab-hevs/magpie`

The medical knowledge is modeled as EC domain dependent axioms that define combinations of events within a time window, which trigger an alert to be notified; and an event is the measurement of a physiological parameter. We consider two different kinds of rules: i) *complex*, where the order of the events is not considered; and ii) *sequential* where the order of the events matters.

3 Demonstration

To demonstrate MAGPIE, we developed an application that tracks the health status of a diabetic patient. In such application, the patient can introduce values of glycemia, weight and blood pressure measurements, as well as the time when the measurements where taken. An agent keeps track of monitoring the patient by reporting alerts related to these three physiological values.

We track glucose rebounds, where blood glucose levels go from low to high in a short period of time, with the following sequential rule,

$$\begin{aligned}&\text{initiatesAt}(\mathit{alert}(p1) = \text{`brittle diabetes'}, T) \leftarrow \\ &\qquad \text{happensAt}(\text{glucose}(V_1), T_1), \text{happensAt}(\text{glucose}(V_2), T_2), \\ &\qquad V_1 \leq 3.8, V_2 \geq 8.0, T_2 > T_1, \\ &\qquad \text{last\_six\_hours}(T_1, T_2), \\ &\qquad \text{not happensAt}(\mathit{alert}(p1), Ta), \\ &\qquad \text{last\_six\_hours}(Ta, T).\end{aligned} \tag{1}$$

The rule above states that a brittle diabetes alert is triggered if two different glucose measurements go from less than or equal to 3.8 mmol/L to more than or equal to 8.0 mmol/L, and three different temporal conditions apply. First, $T_2 > T_1$ specifies the order in which the specified events must happen. Second, the predicate last_six_hours/2 specifies the temporal window in which the events apply. In that case, the predicate checks that the fist event happens no more than six hours before the second event. Third, the last two lines specify the "no alert" condition, which checks that the same alert has not been triggered during the temporal window. This condition avoids possible overwhelming of alerts due to events that make the temporal pattern to hold within the temporal window.

Another sequential rule following the same structure is used to control when the patient is gaining weight, which may indicate that the treatment is not effective and should be revised.

Diabetic patients can also present high blood pressure values, which we track with the following complex rule,

$$\begin{aligned}&\text{initiatesAt}(\mathit{alert}(p3) = \text{`pre-hypertension'}, T) \leftarrow \\ &\qquad \text{not happensAt}(\mathit{alert}(p3), Ta), \\ &\qquad \text{last\_week}(Ta, T), \\ &\qquad \text{more\_or\_equals\_to}(2, (\\ &\qquad\quad \text{happensAt}(\text{blood\_pressure}(\mathit{Sys}, \mathit{Dias}), \mathit{Tev}), \\ &\qquad\quad (120 \leq \mathit{Sys} \leq 139, 80 \leq \mathit{Dias} \leq 89), \\ &\qquad\quad \text{last\_week}(\mathit{Tev}, T))).\end{aligned} \tag{2}$$

This rule states that a pre-hypertension alert is triggered if within the last week, there are two blood pressure readings whose systolic component is within 120 and 139 mmHg and its diastolic component is within 80 and 89 mmHg. The first two lines of the rule specify the "no alert" condition as in the sequential rules. However, for the complex rules it is not feasible to define the other two temporal conditions in the same way that are defined for the sequential rules. The reason is that complex rules do not take into account the ordering of the events. Therefore, a single rule per each temporal permutation of the events should be defined. To deal with that, we define the predicate `more_or_equals_to/2` as follows,

$$\begin{aligned} &\text{more\_or\_equals\_to}(Number, Expr) \leftarrow \\ &\qquad \text{findall}(\_, Expr, List), \\ &\qquad \text{length}(List, Val), \\ &\qquad Val \geq Number. \end{aligned} \tag{3}$$

This predicate counts the number of events in the agent mind that satisfy the conditions defined in its second argument (Expr), and it is evaluated as true if the number of such events is at least equals to the number specified in its first argument.

4 Conclusions

In this demonstration, we show through an Android app how we apply the MAGPIE agent platform to track the health status of a patient affected by diabetes mellitus. Such task is done with temporal rule-based agents that process events related to physiological values of the diabetic patient. Future work in MAGPIE involves its integration with eHealth standards such as FHIR to send the alerts produced by the agents to Tier 3 of the PHS in an interoperable way.

References

1. Brugués, A., Bromuri, S., Pegueroles-Valles, J., Schumacher, M.I.: MAGPIE: an agent platform for the development of mobile applications for pervasive healthcare. In: Proc. of the 3rd International Workshop on Artificial Intelligence and Assistive Medicine. pp. 6–10 (2014)
2. Shanahan, M.: The event calculus explained. In: Artificial Intelligence Today: Recent Trends and Developments, Lecture Notes in Computer Science, vol. 1600, pp. 409–430. Springer Berlin Heidelberg (1999)
3. Touati, F., Tabish, R.: U-healthcare system: State-of-the-art review and challenges. Journal of Medical Systems 37(3), 9949 (2013)
4. Varshney, U.: Pervasive healthcare and wireless health monitoring. Mobile Networks and Applications 12(2), 113–127 (2007)