Journal of Ambient Intelligence and Smart Environments 0 (2013) 1–0
IOS Press

COMMODITY$_{12}$: A Smart e-Health Environment for Diabetes Management

Özgür Kafalı [a,*], Stefano Bromuri [b], Michal Sindlar [c], Tom van der Weide [c], Eduardo Aguilar Pelaez [d], Ulrich Schaechtle [a], Bruno Alves [b], Damien Zufferey [b], Esther Rodriguez-Villegas [d], Michael Ignaz Schumacher [b] and Kostas Stathis [a]

[a] *Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK.*
E-mail: {ozgur.kafali,ulrich.schaechtle,kostas.stathis}@cs.rhul.ac.uk
[b] *Department of Business Information Systems, University of Applied Sciences Western Switzerland 3 Technopole, 3960, Sierre, Switzerland.*
E-mail: {stefano.bromuri,bruno.alves,damien.zufferey,michael.schumacher}@hevs.ch
[c] *Portavita B.V., Oostenburgervoorstraat 100, PO Box 1287 1000 BG, Amsterdam, Netherlands.*
E-mail: {m.sindlar,t.van.der.weide}@portavita.eu
[d] *Electrical and Electronic Engineering Department, Imperial College London, South Kensington Campus, London, SW7 2BT, UK.*
E-mail: {eduardo.aguilar-pelaez02,e.rodriguez}@imperial.ac.uk

Abstract. We present the development of COMMODITY$_{12}$, a Personal Health System (PHS) to assist in the provision of continuous and personalised health services to diabetic patients, thus empowering their lifestyle regardless of their location. COMMODITY$_{12}$ consists of ambient, wearable and portable devices, which acquire, monitor and communicate physiological parameters and other health-related context of an individual, such as physical activity and vital body signals. This data is interpreted by intelligent agents that use expert biomedical knowledge to derive important insights about the individual's health status, which are then presented in the form of active feedback to the patient directly from the device, or via health professionals who assist in diagnosis, treatment and life management. The emphasis of the work is on the design of the PHS in terms of its main components, their integration and deployment to address major problems of interest to both diabetic patients and doctors that treat diabetes.

Keywords: personal health systems, agent technology, diabetes management, interoperability.

1. Introduction

Diabetes mellitus, or simply diabetes, is a group of metabolic diseases in which a person has high blood-sugar, either due to the pancreas failing to produce enough insulin, or because cells do not respond to insulin as expected [2]. Diabetes manifests itself in three types:

- *Type 1*, also referred to as insulin-dependent diabetes, is characterised by the loss of insulin-producing cells in the pancreas which leads to insulin deficiency. It affects 10% of the diabetics in Europe and the US. It can affect both adults and children.
- *Type 2*, also known as non-insulin-dependent diabetes, is caused by insulin resistance, a condition in which cells fail to use insulin properly. Type 2 diabetes is due primarily to lifestyle factors, such as obesity, diet, and

*Corresponding author. E-mail: ozgur.kafali@rhul.ac.uk

sedentary lifestyle, and to a certain extent occurs as a result of genetics. It is the most common type of diabetes, affecting 90% to 95% of diabetics.
- *Gestational diabetes*, a type of diabetes that temporarily affects some otherwise healthy pregnant women. It occurs in about 2% to 5% of all pregnancies and typically disappears after the delivery of the baby. Despite disappearing once the pregnancy is over, it leaves women with a higher risk of later developing type 2 diabetes.

All forms of diabetes increase the risk of short-term complications such as hypoglycemia and hyperglycemia (very low and high blood glucose, respectively), both of which are life-threatening. Longer-term complications include how diabetes affects the eyes (retinopathy), heart (cardiovascular disease), kidneys (nephropathy), and nerves and feet (neuropathy).

The management of diabetes is becoming an increasingly important problem worldwide. In 2000, according to the World Health Organisation, at least 171 million people (or 2.8% of the whole population) suffered from diabetes. The National Health and Nutrition Examination Survey (NHANES III) in the US demonstrated that, in the population over 65 years old, up to 20% have diabetes, with 40% having either diabetes or its precursor form of impaired glucose tolerance. The likelihood of being diagnosed with diabetes increases with age, and the numbers of older persons with diabetes are expected to grow as the elderly population increases in number. It is estimated that by 2030, the number of people suffering from the disease will almost double.

The aim of this work is to present a Personal Health System (PHS), which we will refer to as COMMODITY$_{12}$ [1], seeking to support the management of diabetes. It has been shown that a large portion of the time spent by diabetic patients is devoted to management of self-measured data [51]. The problem here is that despite having a chronic illness, diabetic patients are not able to take a number of crucial decisions related to their treatment, and they need to consult healthcare professionals (such as general practitioners, dietitians, ophthalmologists, chiropodists, pharmacists, specialists and nurses). Thus, one of the main concerns of COMMODITY$_{12}$ is to advise diabetic patients by providing them with recommendations and alerts based on their data, and to assist medical personnel who are in charge of these patients with taking informed and timely decisions. The focus of the work is on diabetes types 1 and 2 in general and on the interaction of these diabetes types with cardiovascular co-morbidities in particular.

We have selected diabetes types 1 and 2 as these two types are chronic conditions that cannot be cured. Type 1 treatment is possible with external supply of insulin and its analogues. Insulin is an injectable drug, and is self-administered by patients using insulin pens. Some patients use continuous insulin injections with miniaturised insulin pumps. Similarly, type 2 diabetes can be controlled with medications. Currently these are mostly oral medications with the addition of some new injectable ones. Their activity is based on increased insulin secretion or sensitisation of the cells toward insulin activity. The important issue here is that all types of diabetes medication share the risk of adverse effects, of which the most serious is hypoglycemia. It is necessary to react to this life-threatening condition immediately, with either drugs or food intake.

The normalisation of the blood-glucose level is one of the parameters that must be monitored and interpreted by a PHS according to a formal model of the disease. Monitoring in our approach means that the system consists of ambient and/or body (wearable and portable) devices, which collect, aggregate and communicate physiological parameters (e.g. blood glucose) and other health-related context of an individual, such as activity and vital body signals. Our approach also encompasses interpretation of these parameters and signals by intelligent agents that use expert biomedical knowledge of the disease to derive important cause-effect relations about the individual's health status. These relations are then presented in the form of active feedback to the patient directly from the device, or via healthcare professionals who assist in diagnosis, treatment and life management.

Our work provides a general agent reasoning framework to draw conclusions from observations. The framework separates domain-specific knowledge (e.g. how to respond to measurement of blood

[1] COMMODITY$_{12}$ stands for COntinuous Multi-parametric and Multi-layered analysis Of DIabetes TYpe 1 & 2.

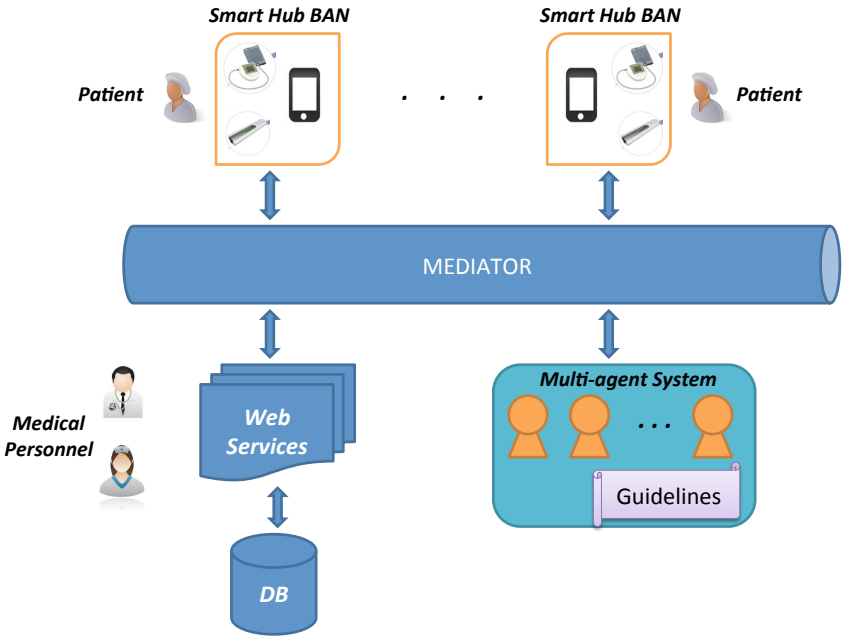


Fig. 1. General architecture of COMMODITY$_{12}$.

glucose) from generic functionality (e.g. diagnosis support) that is reusable in different environments. Also, as different technologies are used in healthcare, our system uses interoperability standards and supports adding new sources of information. To exemplify our work, implementation details are provided wherever necessary, and our system is tested on two important scenarios of diabetes. The specific contributions of our work are:

- We propose a general distributed framework to be used as a PHS for management of diabetes, based on interoperability standards in healthcare informatics.
- We present the body area network of the framework that utilises commercially available sensors to monitor diabetic patients.
- We propose an agent architecture that is founded on model-based diagnosis and supports temporal reasoning about patient observations, utilising formalised medical knowledge, and is able to interact with its environment to communicate results.
- We provide an implementation of the framework, and demonstrate its workings on two important problems related to diabetes.

The rest of this paper is structured as follows. Section 2 introduces the COMMODITY$_{12}$ framework and describes its components. Section 3 introduces the $\mathcal{LAMA}$ agent architecture and explains its most important characteristics. Section 4 reviews our design choices, introduces the agent platform GOLEM used for the deployment of our system, including the details of how interoperability of components is achieved. Section 5 presents our case study for two different scenarios on diabetes monitoring and diagnosis as well as demonstrating different parts of our PHS. Section 6 reviews relevant literature and compares it with our work. Section 7 summarises our approach and outlines our plans for future work.

2. Diabetes Management Framework

To introduce COMMODITY$_{12}$ we first present the general architecture of the system and we briefly describe its main components. We then concentrate on the description of three key components: the commercially available sensors customised for diabetes, the agent-based reasoning

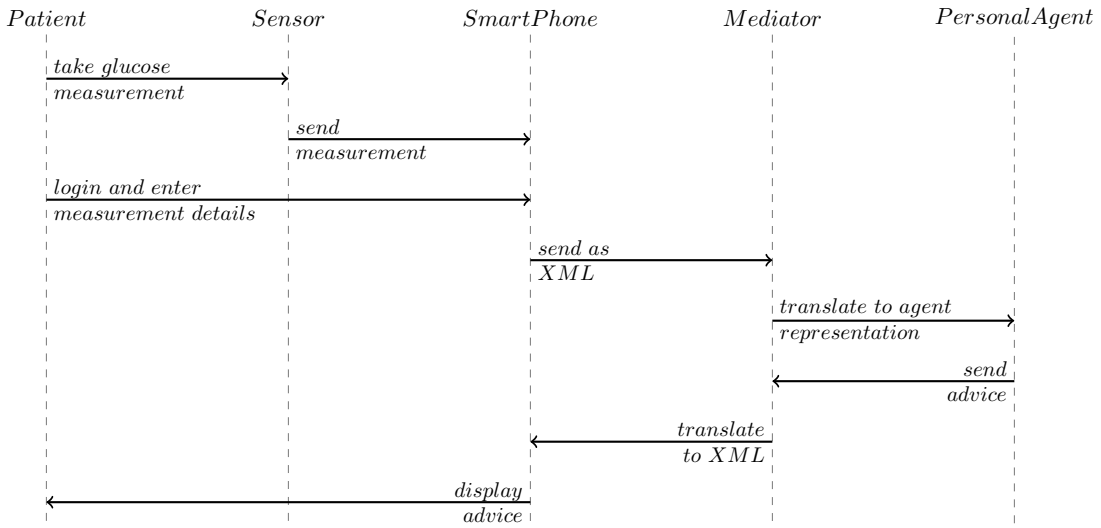


Fig. 2. Sequence diagram for monitoring of a patient.

component, and the infrastructure that enables the interaction among various components in a reliable way.

2.1. PHS Architecture

Fig. 1 depicts our proposed architecture for a smart healthcare environment. It consists of three main components which are connected via the mediator middleware: (i) a smart hub enabling input to be taken directly from the patient, (ii) a system of web services for healthcare personnel, and (iii) a multi-agent system of personal agents associated with patients and medical personnel in order to provide them with advice, alerts and diagnoses.

The smart hub interfaces with the patients in two ways: via the sensors of a body area network (BAN) and via a user interface (UI) on a mobile telecommunication device such as a smart phone or a tablet PC. The web services act as a set of UIs for medical personnel to record observations, and other treatment-related data about the patients. In addition, they provide connectivity with other components of the system through the mediator. The database stores data relevant to the healthcare process for all patients in the system. The personal agents within the multi-agent system are responsible for reasoning based on the observations retrieved from the environment. They make use of logic-based rules that have been formalised out of the medical guidelines selected by the doctors and other domain-specific knowledge related to diabetes.

The components of the framework are connected through the mediator, a middleware for interoperability standards in healthcare informatics. The framework and the architecture will be exemplified by presenting two typical use cases. The first case-study involves a patient monitoring his blood-glucose level using the system (see Section 5.1), as shown in the sequence diagram of Fig. 2. The second case-study is concerned with the examination of a patient's diabetic foot by a doctor (see Section 5.2), as shown in the sequence diagram of Fig. 3. The details of how data flows through the middleware (e.g., translations between components) will be described in Section 2.4.

2.2. Sensors for Diabetes Monitoring

We utilise various types of sensors in our PHS, which can be divided into different categories based on their suitability for inclusion in the present smart hub. The sensors are depicted in Table 1 and described below, with the objective of seeing whether they meet the requirements for our purpose.

The following wireless sensors are available on the market, and are already capable of connecting to the smart hub:

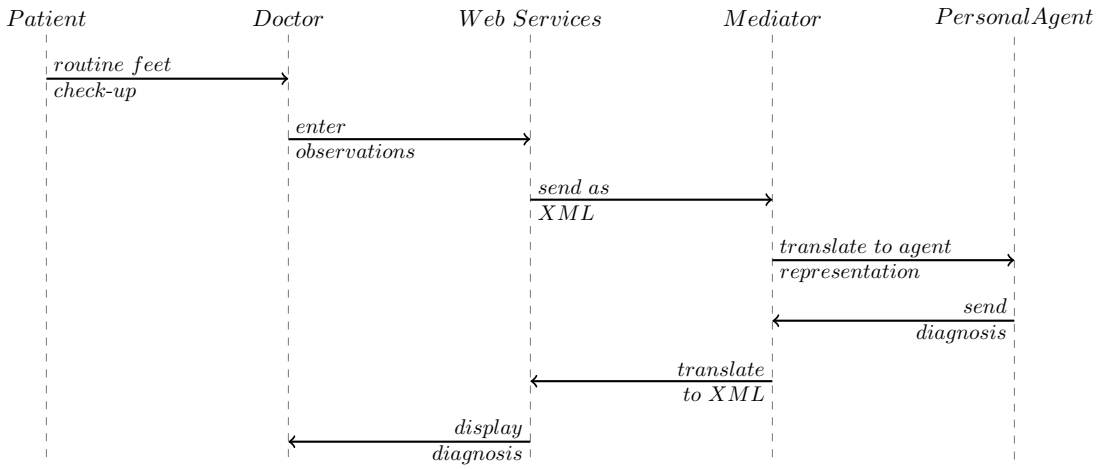


Fig. 3. Sequence diagram for diagnosis of a patient.

– Glucose monitoring by finger prick (GlucoTel sensor[2], Table 1a).
– Blood pressure monitoring, non-continuous (PressureTel sensor[2], Table 1b).
– Body weight monitoring (WeightTel sensor[2], Table 1c).

None of the above sensors is expected to be considered obtrusive by the patients. In addition to these sensors, we consider adding the following sensors to the smart hub, all of which are acceptable for the patient to wear during the day/night:

– Respiratory effort monitoring (thoracic band tension sensor[3], Table 1d). This sensor is of a similar size and weight to widely popular sports heart rate monitors.
– Activity monitoring (tri-axial accelerometer[3], Table 1e). This sensor is planned to be mounted on the chest band of the respiratory effort sensor. Also, additional mounting points are possible in order to gather alternative information about the patient's activity. Activity monitoring will be done in a multi-parametric fashion including; fundamental signal pre-processing, artefact detection, feature extraction and inactivity detection for data rate reduction to be done at either the sensors of the smart phone application. Further interpretation will be done at later stages of the system in combination with other activity-related contextual data (such as smart phone accelerometer activity and user input) where the agents will work towards diagnosis assistance. In the remit of accelerometer signals interpretation one avenue to be explored is energy expenditure estimation where [39] represents a strong starting point.
– 3-channel ECG (available soon from BodyTel in collaboration with IEM[4], Table 1f). The main advantage of this sensor is its similarity to Holter monitors, which are commonly used for ambulatory cardiovascular monitoring.

The first two sensors described above require electronic signal conditioning and processing as well as wireless capability to be added. We consider doing this using low-power MPS430 micro controllers [62] and the SimpliciTI [61] wireless protocol by Texas Instruments, in combination with the low-power nRF24LE1 Nordic Transceiver [44].

Moreover, the following sensor is considered for clinical trials, but not included in the BAN:

– Continuous glucose monitor (CGM[5], Table 1g). A possible application of this sensor is to educate patients on the fluctuations of blood glucose levels in their body. This sensor is to be partially implanted on the side of the abdomen and to be worn for 3 to 5 days con-

[2] http://www.bodytel.com
[3] http://www.biopac.com
[4] http://www.iem.de/beam
[5] http://www.medtronicdiabetes.com/products/guardiancgm

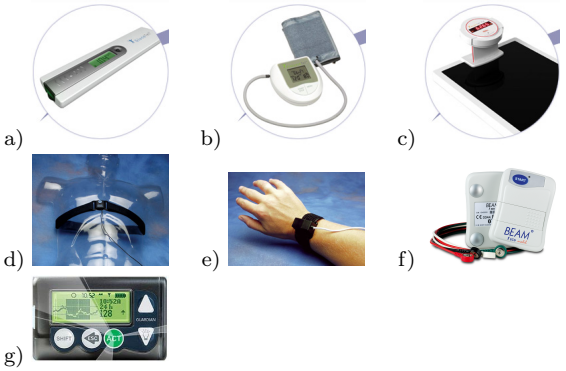


Table 1
Sensors (a, b, c, and f by BodyTel, d and e by BioPac, g by MedTronic): a) GlucoTel sensor. b) PressureTel sensor. c) WeightTel sensor. d) Thoracic band tension sensor. e) Tri-axial accelerometer. f) 3-channel ECG. g) Guardian continuous glucose monitor.

tinuously; hence, it is expected that some initial reticence and potential discomfort will be incurred to the patient.

The sensors described above offer great potential for intelligent data acquisition based on real-time data streams. Given their properties as described above, they are considered suitable for monitoring patients with Type 1 and Type 2 diabetes, and to be utilised in our PHS.

2.3. Personal Agents and their Deployment

The COMMODITY$_{12}$ PHS uses personal agents as active software components to (a) manage the specific characteristics of individuals in user profiles, (b) become aware of contextual conditions about the domain by observing the environment through sensors, and (c) process intelligently information from observations by reasoning logically with them to support monitoring, the provision of alerts, advice, and more sophisticated tasks such as diagnosis. Intelligent processing here is construed as the ability of the agent to operate with the complexity that arises from the environment and the ability of the agent to take action in the presence of incomplete information.

Our work proposes the $\mathcal{LAMA}$ (Logic Agents for Medical Advice) agent architecture, which will be used to build COMMODITY$_{12}$ agents in general and develop their reasoning capabilities in particular. The reasoning capabilities will allow a personal agent identify time-critical observations and react to them without ignoring goals and plans the agent has developed during the interaction with the patient or the relevant medical personnel.

To deploy multiple $\mathcal{LAMA}$ agents we use the agent platform GOLEM [13,14]. This platform has been selected because it has been developed specifically for the deployment of logic-based agents, like $\mathcal{LAMA}$, and has been used successfully in a number of applications [63,15,16,12,42]. GOLEM also supports inter-agent communication over a network and the deployment of objects (entities that are not agents) as well as mechanisms of interaction between agents and objects. Moreover, GOLEM uses the notion of container to deploy agents on different machines over a network, thus giving the possibility to develop complex distributed applications. Next, we focus on the infrastructure of our framework. More details on $\mathcal{LAMA}$ will be given in Section 3.

2.4. The Mediator

One of the main issues with systems developed from different and heterogeneous components is how to handle component interoperability. In COMMODITY$_{12}$ interoperation of this kind is particularly important as the GOLEM platform uses first-order logic terms to represent information while sensors and web-service do not. Therefore we need to transform data such as physiological values coming from sensors or patient information coming from the medical database into logical terms accepted by GOLEM and vice-versa.

The integration problem in medical systems is well documented, for example see [10]. Heterogeneity has many consequences on the development of interoperable systems and has driven the community to develop a new kind of software technology: the Enterprise Service Bus (ESB). Instead of developing countless adapters to bridge the differences in terms of formats and protocols, the ESB approach provides a complete centralised and streamlined message transformation framework. The responsibility of interpreting the data is thus shifted towards the ESB and is done in a single place.

To support the COMMODITY$_{12}$ mediator, we employ Mirth Connect (MC) [6], an ESB interop-

[6] http://www.mirthcorp.com/products/mirth-connect

erable communication framework between healthcare systems that supports a wide range of data standards and communication protocols, such as HL7 v2/v3, DICOM, HTTP, MLLP, databases and web services. Communication in MC is organised in channels. Data is channelled from one side of the pipe; it is then filtered, transformed and then dispatched towards its final destination. Each channel receives data through one single endpoint, called the source connector. This connector accepts requests (or creates requests) from (to) external systems and provides support for several protocols (Database, File, HTTP, MLLP) and formats (HL7 v2/v3, XML, X12, DICOM). The filtering is done through either static rules or E4X scripts (XML-support for e.g. JavaScript). Transformations are applied on the message level using XSLT templates, static mappings, or JavaScript. Finally the outbound data is forwarded to its target system through one or more destination connectors.

Coupling MC and GOLEM in COMMODITY$_{12}$ allow us to interoperate the GOLEM agent technology with a large set of medical devices and systems. In this way we can deploy personal agents in the PHS to provide the required high-level and logic-based functionality for monitoring, alerts, advice and diagnoses.

3. The $\mathcal{LAMA}$ Agent Architecture

This section introduces the $\mathcal{LAMA}$ model of agency, a computational logic architecture that integrates reactivity and goal-driven planning in the environment in which the agent is situated. $\mathcal{LAMA}$ is a cut-down version and, in some cases, an extension of the KGP (Knowledge, Goals, Plans) agent model [28] with specific focus on the medical domain. KGP allows the independent specifications of a collection of computational logic capabilities especially suited for agents situated in open and dynamic environments, where the knowledge about the environment is usually incomplete.

3.1. Overview

Fig. 4 demonstrates the $\mathcal{LAMA}$ agent architecture. As in [56,13], the agent is composed of two main software components, a *body* and a *mind*. The body is responsible for physical capabilities

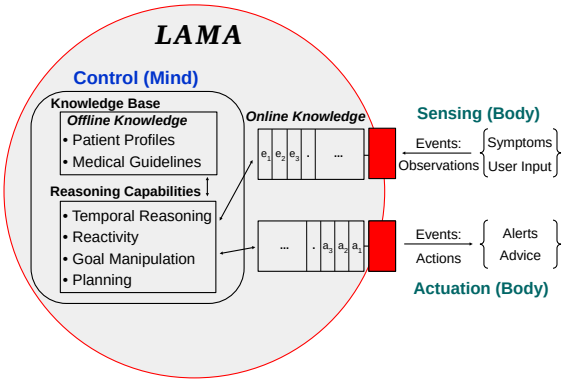


Fig. 4. $\mathcal{LAMA}$ agent architecture.

like sensing and actuation, thus situating the agent in an environment. The mind, on the other hand, allows the agent to symbolically process information coming from sensors using the reasoning capabilities of the agent. During the execution of the agent, the body perceives the agent's environment, and if it senses a change, it will ask the mind to revise its current action to reflect the perceived environment changes. The mind also controls the agent's knowledge base, which contains medical knowledge about the domain.

First, we look at how knowledge is represented in $\mathcal{LAMA}$. We differentiate between *online* and *offline* knowledge.

- *Online knowledge* represents dynamic data that is updated through continuous perception of the agent. These can come in as data entered from a user interface of the system (e.g., the physician recording observations of the patient) or imported automatically (e.g. laboratory results).
- *Offline knowledge* represents the profile data for the patient and disease-related facts and rules, which will be updated occasionally. These include static facts related to patient's profile (e.g., patient's birth date), the medical expertise and guidelines that are required to draw conclusions for given situations.

Next, we review the physical capabilities of $\mathcal{LAMA}$.

- *Sensing* is a physical capability of the body that allows the agent to observe that properties hold or do not hold, and that events have occurred in the environment.

– *Actuation* is another physical capability that allows the agent to act on its environment through both physical and communicative actions.

Furthermore, we review the following reasoning capabilities of $\mathcal{LAMA}$.

– *Temporal reasoning* is the capability that allows the agent to reason with temporal information. It is invoked by other components of the agent, to prove or disprove that a fluent literal holds, with respect to a given corpus of known facts and events.
– *Reactivity* is the capability that allows the agent to react to stimuli from the external environment. When the body of the agent perceives an observation that triggers a reaction, the mind must respond immediately, possibly by suspending current goals and activities that can wait.
– *Goal manipulation* is the capability that allows the agent to decide, at a given time point, which goals are to be pursued, and which goals are obsolete. For the ones that are still pursued, the agent will then select actions aiming to achieve them.
– *Planning* is the capability of finding a sequence of sub-goals and actions in order to achieve a goal. Here, we assume that all goals are already planned for, and the agent uses a predefined plan library.

3.2. Reasoning

The idea behind the reasoning capabilities of $\mathcal{LAMA}$ is to create a practical reasoning framework that is rich enough to express knowledge representing guidelines and medical expertise, use this knowledge to diagnose conditions, monitor plans and take action to provide advice and alerts to patients and medical personnel. We have chosen Prolog to be the basis of agent reasoning, because of its unification capability, proof-oriented computation, suitability for meta-level reasoning and wide availability of tools. Prolog, for instance, makes it possible to compute justifications of generated advice, which is important if such advice is to be accepted by medical practitioners. Here, a meta-logic approach [32] is used to incorporate the representation power of extended logic programs in the framework[7].

The framework is based on a hierarchical organisation of conclusions. The most basic level of conclusions in the framework are the ones that hold because the agent has perceived them via sensing the environment. More specifically, we treat the observations and actions executed by the agent as events that happen in time, using the temporal reasoning capability. As in the original KGP model, a meta-logic approach allows us to refer to object-level concepts such as the *state* of the agent, *fluents* representing properties in the state of the agent that change over time and events that happen at a time. The underlying time model that we assume is linear and may include real numbers or integers. Temporal intervals will be represented as pairs of time-points.

```
holds_at(P, T):-
   T=0,
   initially(P).

holds_at(P, T):-
   greater_than(T, 0),
   holds_for(P, [Start, End]),
   greater_than(T, Start),
   less_than(T, End).

holds_for(P, [Start, End]):-
   initiates_at(Ei, P, Start),
   terminates_at(Et, P, End),
   greater_than(End, Start),
   \+ broken_during(P, [Start, End]).

holds_for(P, [Start, infPlus]):-
   initiates_at(Ei, P, Start),
   \+ broken_during(P, [Start, infPlus]).

holds_for(P, [infMinus, End]):-
   terminates_at(Et, P, End),
   \+ broken_during(P, [infMinus, End]).

broken_during(P, [Start, End]):-
   (
    terminates_at(E, P, T);
    initiates_at(E, P, T)
   ),
   less_than(Start, T),
   greater_than(End, T).

terminates_at(E, F = V, T):-
   initiates_at(E, F = Vnew, T),
   holds_at(F = V, T),
   V \= Vnew.
```

Listing 1: Domain-independent axioms of Event Calculus.

[7]Throughout this section we assume familiarity with logic programming and Prolog.

We use the Event Calculus (EC) to develop the temporal reasoning capability of $\mathcal{LAMA}$ agents. The EC is a logic programming formalism for reasoning about actions and their effects [33,54]. The dialect of the EC adopted here assumes multivalued fluents, where a fluent is represented in the form $F = V$ to denote F has value V [7]. In this dialect *boolean* fluents are a special case in which the possible values are `true` and `false`. For the top-level specification of the EC implementation, we use an efficient version with maximal validity intervals and multi-valued fluents. The domain-independent axioms are given in Listing 1.

We give next an example of temporal reasoning that relates to diabetes:

```
initially(calves_pain(right)=false).
initially(calves_pain(left)=false).

happens(complains_of(calves_pain, right), 1).

initiates_at(
   complains_of(calves_pain, S),
   calves_pain(S)=true,
   T):-
   happens_at(complains_of(calves_pain, S), T).
```

Listing 2: Agent's knowledge about the patient.

This domain-specific information states that initially the agent believes that a patient does not have pain in either of his calves. However, after observation at time point 1 the patient complaint about pain in the right calf muscle, the observation is asserted as an event that happens. Then an `initiates/3` rule states that the fluent corresponding to the agent's belief that the patient has pain on calf muscle side S is initiated, if a calf pain event for foot side S has happened. The query

`holds_at(calves_pain(right)=true,1)`

is answered by the agent through the use of the domain-independent EC axioms together with domain-specific knowledge, and will reply negatively because `calves_pain(right)=true` has not yet been initiated at $T = 1$. However, if we ask the same query at time 2, then the agent will reply affirmatively, since 2 is a time after the observation that the patient complained about pain in her calf.

At the next level of the reasoning framework we can support deduction with complete knowledge by describing how the agent can draw *necessary conclusions* using domain rules whose conditions are completely provable (in the classical sense) at that point within the knowledge base of the agent. Conditions in the body of such rules can be:

- explicit prior knowledge that holds at all times, e.g., profile information such as the ethnicity of a patient, the fact that a patient is diabetic, or the target range/type of a value;
- fluents whose values hold at a particular time because the agent observed events that initiated or terminated them, e.g., that a patient suffers from calves pain on the right, or that she has been prescribed a particular kind of medication;
- new necessary fluents derived as a result of other necessary conclusions.

As an example of necessary conclusions and their supporting knowledge is given below:

```
necessary_at(diabetic(Patient)=true, T):-
   necessary_at(diabetic(Patient, Type)=true, T),
   member(Type, [type1, type2]).
necessary_at(P=V, T):-
   holds_at(P=V, T).

happens_at(diagnosis(p145, diabetic(type2)), 2).

initiates_at(
   diagnosis(Patient, diabetic(Type)),
   diabetic(Patient, Type)=true,
   T):-
happens_at(diagnosis(Patient,diabetic(Type)), T).
```

Listing 3: Drawing necessary conclusions.

The first rule expresses the fact that a patient is diabetic if the patient has been diagnosed with type 1 or 2 diabetes. To prove that a patient (say patient $p145$) has been diagnosed diabetic (say of type1) at a specific time (say at time 3), can be done via the second rule that calls the temporal reasoning via the EC `holds_at/2` that uses the event information and the properties it initiates. In this way the agent uses necessary rules for fluents sensed in the environment to draw conclusions about the patient.

In many situations an agent may have incomplete knowledge about the conditions that will make the agent establish that a conclusion G as true (or false) at a time T. This problem often arises because the agent is either unable to observe the truth value of the conditions in the environment, or because it lacks rules that derive these conditions from the current state of its knowledge base. To deal with this, we allow agents to draw

possible conclusions about conditions that are unknown to them, as if these conclusions were hypothesised.

As an example of possible conclusions, consider the following rules:

```
possible_at(feet_Simms_class(3)=true, T):-
   possible_at(amputated(F)=true, T),
   foot(F).

possible_at(feet_Simms_class(3)=true, T):-
   possible_at(ulcer(F)=true, T),
   foot(F).

foot(left_foot).
foot(right_foot).
```

Listing 4: Drawing possible conclusions.

The rules above follow the Simm's classification guidelines for determining the foot risk category for diabetics (see Section 5.2), which state that a patient belongs to risk class 3 if he/she has an amputated foot or has an ulcer on one of the feet. Typically, this classification is used deductively, to conclude a certain risk class based on *known* facts. If certain fluents are unknown, however, reasoning with possible conclusions hypothesises specific values to boolean fluents, thus helping doctors to decide which additional observations to make for a specific case. As with [27], our reasoning process keeps track of the set of hypotheses that have been made to prove a conclusion. A fluent is hypothesised if it is undefined in the current state of the agent and its hypothesis is consistent with the hypotheses made so far, as if they are satisfying the domain-specific integrity constraints.

In order to deal with the problem of integrity in necessary and possible conclusions, we need to filter out boolean fluents that are `true` and `false` at the same time. To address this problem we define *legal conclusions* that can be thought of as specifying an integrity checking process in the knowledge base of the agent for necessary and possible conclusions. Below we give an example of how these constraints are defined for necessary conclusions:

```
necessarily_at(P=V, T):-
   necessary_at(P=V, T),
   legal_at(necessarily(P=V), T).

legal_at(necessarily(P=V), T):-
   \+ boolean(V),
   in_range(P=V, T).
legal_at(necessarily(P=V), T):-
   boolean(V),
   opposite(V, OppV),
   \+ necessary_at(P=OppV, T).

boolean(true).    opposite(true, false).
boolean(false).   opposite(false, true).
```

Listing 5: Necessary conclusions that are legal.

The first rule defines necessary conclusions that are legal. The second rule states that, for non-boolean fluents, a necessary conclusion is legal if the value of the fluent is in the expected range, a condition that is application defined. Then the third rule checks that, for boolean fluents, the explicit opposite value is not necessary at the same time, thus eliminating any inconsistencies. We can define possible conclusions similarly:

```
possibly_at(P=V, T):-
   possible_at(P=V, T),
   legal_at(possibly(P=V), T).

legal_at(possibly(P=V), T):-
   \+ boolean(V),
   in_range(P=V, T).
legal_at(possibly(P=V), T):-
   boolean(V),
   opposite(V, OppV),
   \+ possible_at(P=OppV, T).

possible_at(P=V, T):- necessarily_at(P=V, T).
```

Listing 6: Possible conclusions that are legal.

The first rule now defines possible conclusions that are legal, while the second and third rules deal with the range of values and inconsistencies in possible conclusions. The final rule shows how legally necessary conclusions persist at the possible conclusions level, thus establishing the missing link between necessary and possible conclusions.

Although legal conclusions allow us to use hypothetical reasoning whose conclusions are consistent, it does not allow us to express preferences over legally possible conclusions. For example, in the diabetic foot domain, when there is loss of protective sensibility in one foot then the patient may belong to class 2 if that patient has in addition an

arterial condition. However, if the patient has loss of protective sensibility only, then she belongs to class 1. So if the patient belongs to class 2 following the guidelines, then the patient belongs to class 1 too. In this case, the guidelines suggest, that advice should be provided by taking the higher class into account.

In practice, the problem described above is often solved in Prolog systems using the cut (!) operator. However, this operator makes the rules not declarative. To avoid introducing cut into the rules of our framework, we define *choices* over derived fluents with similar legal conclusions. These choices are further defined in terms of *priorities* that allow us to choose the conclusion with higher priority. Then we extend legally possible conclusions with what we call *valid conclusions* as follows:

```
valid_at(P=V, T):-
   possibly_at(P=V, T),
   \+ invalid_at(P=V, T).

invalid_at(P=true, T):-
   is_type(choice, P),
   higher_priority(Q, P),
   possibly_at(Q=true, T).
```

Listing 7: Drawing valid conclusions.

Legally possible conclusions that are invalid are filtered out via the first rule. For choices, a fluent that is possibly true is invalid if there is higher priority conclusion that is also possibly true at the same time. In this way, we can avoid using non-logical primitives in our rules, and thus having a reasoning framework that is expressive, declarative and extensible.

3.3. Control

We can now build the agent that uses rules of the kind described above to provide advice. The control component of the agent to realise this purpose is provided by a reasoning cycle [26] that is executed every time the body performs a perception on the environment and calls the mind. The body calls the mind with the query:

```
?- cycle_step(State, Action, NewState).
```

`State` is a term of the form:

```
(Time, Observations, Goals)
```

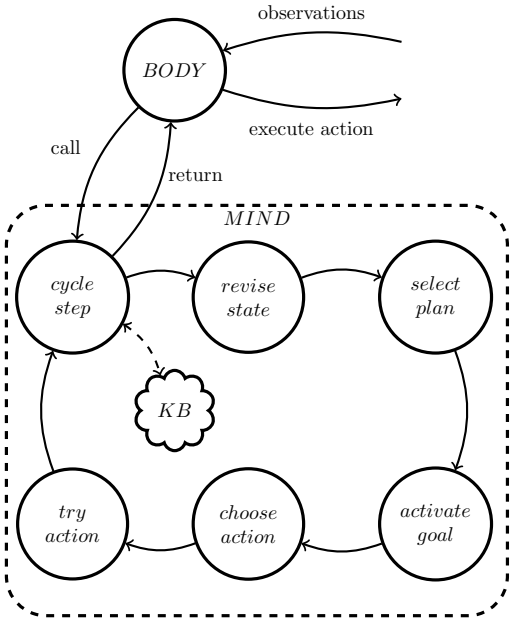


Fig. 5. $\mathcal{LAMA}$ agent cycle.

containing the `Observations` and `Goals` of the agent at the current `Time`. It is provided as input to the agent cycle that in one step produces an `Action` to be performed by the body and a `NewState`. In a single cycle step, the mind performs the following tasks:

1. Revises the internal state according to the new observations coming from the body,
2. Selects a plan for the revised state, creating new goals.
3. Activates a goal from the set of new goals.
4. Chooses an action in order to achieve the activated goal.
5. Tries the selected action (if the action has a mental part like adding a sub-goal), and returns it to the body together with the newly formed state of the agent.

This agent cycle is summarised in Fig. 5, and it can be described in Prolog as follows:

```
cycle_step(State, Action, NewState):-
   revise(State, RevisedState),
   plan(RevisedState, NewGoals),
   activate(NewGoals, Goal),
   choose(RevisedState, Goal, Action),
   try(RevisedState, Goal, Action, NewState).
```

Listing 8: The agent cycle in Prolog.

After the cycle is completed, the body executes the chosen action in the environment, e.g., give advice to the doctor. This action is selected as part of a plan that is aimed towards the achievement of a specific goal. We distinguish between two types of goals, achievement and maintenance goals [65,25]. An achievement goal corresponds to a condition that the agent needs to satisfy once within a given period of time, e.g., scheduling of the next foot examination of the patient. A maintenance goal, on the other hand, corresponds to maintaining a condition for a period of time. For example, if the agent's goal is to monitor the patient and make sure his blood sugar does not drop to a critical level, then it will succeed in that goal if the level of the patient's blood sugar stays within the normal limits for the monitoring period.

4. GOLEM and Mediator Functions

In our PHS, sensor devices stream off physiological values to a centralized care management system (CMS). The mediator translates these values in the right formats for all the components involved. Once, the smart hub submits to the mediator the physiological data of the patient, we use Javascript on the MC side to translate the messages exchanged in the platform into HL7. Example messages that the smart hub sends include the following observations: (i) glucose, (ii) blood pressure, (iii) weight, and (iv) pulse.

Fig. 1 shows the data flows taking place in the PHS. All the information produced by the BAN is translated to an HL7 CDA document to be stored in the CMS. Furthermore, the stream of data is also translated to logical terms of GOLEM. Every time the smart hub connects to MC to send the physiological values, MC translates those into EC events that $\mathcal{LAMA}$ agents can interpret. Such events are then submitted to GOLEM which notifies the agent subscribed to the events associated to a certain patient identifier, see Listing 9, line 1, for an example.

In this sense, every agent has a set of sensors to perceive such events and GOLEM notifies the agents that are subscribed to the events produced regarding a certain patient. Namely, we can specify the subscription of an agent sensor to glucose events as given in Listing 9, lines 2-3. This means that an agent owns sensor s1 that listens to blood_pressure events of patient p1. Similarly, we can specify a sensor that listens to glucose events as given in Listing 9, lines 4-5.

```
1 happens(glucose(p1, 5.5, breakfast, 197), 200).
2 happens(sensor(s1, blood_pressure(
3    p1, Systolic, Diastolic, Time), 200)).
4 happens(sensor(s2, glucose(
5    p1, Glycemia, Period, Time)), 201).
6 happens(retrieve_examinations(ag1, p1), 200).
7 happens(retrieve_lab_tests(ag1, p1), 200).
```

Listing 9: Events in GOLEM.

Below we show part of a sample HL7 CDA document body for our PHS, to handle the blood pressure and glucose observations when they are submitted by MC to the care management system. An HL7 document is composed of a header, a body and a footer. Here, we will only discuss the content of the body, as this is the part where we specify our observations.

```
<observation classCode="OBS" moodCode="EVN">
<templateId root="2.16.840.1.113883.10.20.1.31"/>
<id root="107c2dc0-67a5-11db-bd13-0800200c9a66"/>
<code code="271649006"
   codeSystem="2.16.840.1.113883.6.96"
   displayName="Systolic blood pressure"/>
<statusCode code="completed"/>
<effectiveTime value="200003231430"/>
<value xsi:type="PQ" value="132" unit="mmHg"/>
<interpretationCode code="N"
   codeSystem="2.16.840.1.113883.5.83"/>
<referenceRange>
<observationRange>
   <text>N 100-135 mmHg</text>
</observationRange>
</referenceRange>
</observation>
```

Listing 10: CDA HL7 message for blood pressure.

An observation OBS is a subclass of an act of type event EVN. An act is any kind of medical act during the check-up in the visit to the doctor. The template ID 2.16.840.1.113883.10.20.1.31 is a pointer to the template describing the content of the observation. The id root 107c2dc0-67a5-11db-bd13-0800200c9a66 identifies uniquely the observation. The message contains a single observation "Systolic blood pressure", SNOMED code 271649006, with numeric value 132 mmHg. The code system id 2.16.840.1.113883.6.96 identifies the SNOMED CT code system. The effective time is the time stamp of the event, while PQ

```
1 possible_at(select(monitor_hypoglycemia, alert([doctor], severe_hypoglycemia))=true, T):-
2     lastday(Ts, T),
3     findall(Val, (holds_for(glucose=value(Val), [Ts, T]), severe_hypoglycemia(Val)), HypoCases),
4     length(HypoCases, L), L >= 2.

6 severe_hypoglycemia(Value):- Value >= 3, Value =< 4.
```

Listing 11: Possibility rule for severe hypoglycemia alert.

stands for Physical Quantity and it refers to the value and the unit. The interpretation code defines if the observation is normal, high or low; in this case normal. The observation range is a text comment specifying the range of the values for being in normal condition.

In COMMODITY$_{12}$ agents can also query the patient's profile. In particular, the profile contains: (i) physical examinations like the annual foot examination, the eye check, the check for neurophatic symptoms, waist circumference, (ii) lab tests like cholesterol values, the HbA1c value, blood pressure, serum values, protein in the urine, and (iii) the patient's history like family history and habits, such as smoking and drinking alcohol.

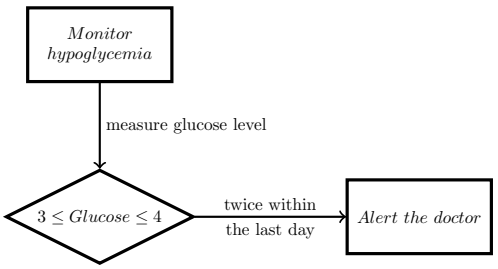
To achieve such a communication between GOLEM and PHS DB (as depicted in Fig. 1), we define a set of queries in both GOLEM and in the DB. In GOLEM, we specified a set of events in the EC formalism to deal with the notification and subscription of lab test events and physical examination events to the $\mathcal{LAMA}$ agents. Every time the agent is deployed in GOLEM to serve its own patient, the agent requests the changes in the patient profile, by requiring the most recent physical examinations and lab tests. To achieve this, after deployment time, the agent produces the events given in Listing 9, lines 6-7.

Such events are submitted to MC by GOLEM, which then submits the query to the DB. The DB queries the repositories by means of SQL statements. The result of the SQL query is returned back to MC in HL7 format and translated to predicates. We use a DB schema based on the HL7 V3 RIM model [52,24]. The use of HL7 V3 RIM is fundamental as this simplifies the creation of messages that respect the HL7 CDA format.

5. Case Study

5.1. Hypoglycemia Monitoring

Hypoglycemia is the case where the glucose level of the patient is below a certain threshold value (International Classification of Diseases (ICD): ICD-10, E16.0-E16.2)[8]. Continuous monitoring of the glucose level for different periods of time (e.g., before lunch or before dinner) is essential in detecting an abnormal case for the patient. According to the severity level of hypoglycemia, different actions may need to be taken including alerting the doctor. Here, we focus on one specific case that is described by the following algorithm:



In particular, the rule we are interested in describes a severe hypoglycemia case, where the patient has two cases of hypoglycemia within the last day. In such a case, the action to be taken is to alert the doctor of this situation. Listing 11 demonstrates how we represent this rule in our reasoning framework. This rule is stored in the agent's knowledge base where it can be accessed at every cycle of the agent's execution. The head of the rule selects a suitable action for the given situation given that the conditions in the body of the rule hold. So, for our specific rule, when the current situation of the agent is to monitor for hypoglycemia (line 1), then in order for the rule to fire, the following needs to hold:

[8] http://www.who.int/classifications/icd/en

– The time period to be checked is within one day (line 2).
– At least two cases of severe hypoglycemia have occurred within the last day. Using temporal reasoning, line 3 finds all glucose measurements from the last day that classify as severe hypoglycemia. Line 4 expresses that at least two of such measurements must be found. Finally, line 6 states that a glucose value is classified as severe hypoglycemia when it is between 3 and 4 mmol/l.

Note that glucose measurements can be provided by the patient via the smart hub as well as by the doctor through the web interface. When the measured glucose values are at critical levels, the agent selects the action to alert the doctor (line 1). This alert is then passed to the doctor via the connection between GOLEM - Mirth Connect and Mirth Connect - web interface. The recipient of the alert contained within the action description.

5.2. Foot Ulceration

Foot complications are among the most serious and costly complications of diabetes, possibly resulting in the amputation of the patient's lower extremity. Because of the nature of the *diabetic foot*[9], a multidisciplinary approach is needed to prevent ulceration, amputation, decrease in patients' quality of life, and high medical costs. Early detection of the at-risk foot, as well as adequate and timely treatment, are key aspects of such an approach.

In order to aid medical practitioners, general guidelines have been developed that allow for classification of diabetic patients into different risk categories, based on the condition of their feet [5]. Typically, those guidelines describe basic principles for prevention and treatment of the diabetic foot, and need to be translated for local use when implemented in practice, taking into account regional differences in socio-economics, accessibility to healthcare, and cultural factors.

In the Netherlands, the Dutch Association of Internists has put forward *Guidelines for the Diabetic Foot* describing four different risk categories related to the diabetic foot, each of which has an associated check-up frequency with which the specialist is advised to see the patient. These guidelines encompass a set of rules known as the "modified Simm's classification" [64,55], as shown in Table 2.

Class	Risk profile	Frequency
0	No LPS or PAD	1x / 12 mo
1	LPS or PAD, without LIP	1x / 6 mo
2	LPS together with PAD or LIP	1x / 3 mo
3	History of ulcers or amputation	1x / 1–3 mo

Table 2
The modified Simm's classification.

According to the rules of the modified Simm's classification, the patient falls into risk category 2 when she shows loss of protective sensibility (LPS) in the feet, together with peripheral arterial disease (PAD) or signs of locally increased pressure (LIP). The physician is then advised to see the patient at least once every six months. To determine if the patient is afflicted by any of the factors that influence the outcome of the Simm's classification, the physician performs an anamnesis and inspects the patient's feet. The data is then entered into the PHS, using a form as in Fig. 6 that depicts part of the entry form for the foot examination. Whether or not the patient has a history of ulcers or amputation is observed by the physician during the examination, or inferred from historical observations available to the PHS. Other observations that are related to the presence of items in the Simm's risk profile of Table 2, are as follows:

LIP is indicated if the patient has any of:

– tylosis or clavus
– pressure sores
– a skin defect, or signs of infection

LPS is indicated if the patient has any of:

– superficial sensitivity disorder, as measured using the Semmes-Weinstein monofilament

PAD is indicated if the patient has any of:

– pain in calves while walking
– spasmodic pain in calves while lying down
– difference in temperature between his/her feet
– no pulsation in a single *a. tibialis posterior*
– no pulsation in a single *a. dorsalis pedis*
– no pulsation in both *a. tibialis posterior*, and no pulsation in both *a. dorsalis pedis*

[9] The term "diabetic foot" here refers to the spectrum of foot deformations as a result of neuropathy, macroangiopathy, limited joint mobility, and metabolic disorders, which typically occur in diabetic patients [5,66].

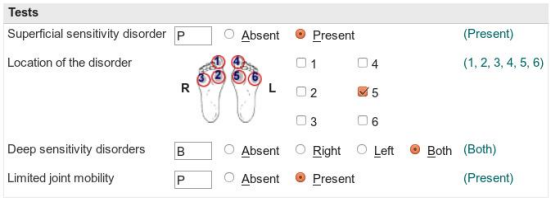


Fig. 6. Foot examination entry form.

The rules of the Simm's classification are formalized in Listing 12. Let us now look at an interesting case when a given observation triggers both risk categories 1 and 2. The conditions needed to conclude that a patient belongs to category 1 and category 2 are given in the listing. Note that the observation *loss_of_protective_sensibility*, which is a *neuropathy_condition*, is included in both classes. In order to resolve conflicts that may arise from observations of this type, we use valid conclusions as described in Section 3. So, if it is possible to conclude both category 1 and category 2 out of the observations for the patient, then only category 2 will be valid since it has higher priority.

For example, a doctor first fills in the observations about the patient, using the web-based interface (Fig. 6) of the Portavita PHS that is used as basis for the COMMODITY$_{12}$ architecture. To simulate the case described above, the correct parameters are entered using the form elements of the interface. Then, based on the rules of the classification, it is possible to conclude both Class 1 and Class 2. Finally, Class 2 is selected as the final Simm's classification category, advising a foot examination once every three months. The resulting advice is depicted in Fig. 7.

5.3. PHS User Interface

The treatment of diabetes involves caregivers from multiple disciplines, such as general practitioners (GPs), dietitians, ophthalmologists, chiropodists, pharmacists, specialists and nurses. The set of Web services developed in the Portavita[10] system helps caregivers managing the care for chronic diseases such as diabetes. All caregivers in the treatment of a patient can be authorized to parts of the patient health record such that they can give a specific contribution. Moreover, the patient can be actively involved because she

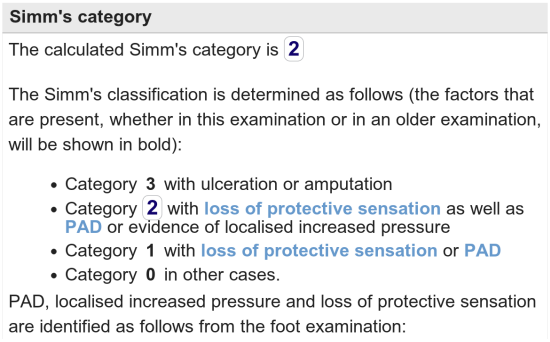


Fig. 7. Generated advice for the foot exam.

has access to her health record and can perform self-examinations. The Portavita Diabetes KIS was produced in close cooperation with GPs, internists, paramedics, nursing staff and the patients' association DVN[11]. It was also based on the guidelines of the Dutch diabetes foundation[12].

After the patient is diagnosed with diabetes, the treating physician asks for the disease history of the patient, lifestyle and other general conditions. The patient and the physician together then use this information to construct an individual treatment plan. The treatment plan contains target values concerning body weight, glucose regulation, blood pressure, lipids and kidney function. Furthermore, the physician and patient agree upon the monitoring of the patient's adaptation of lifestyle, cardiovascular risk profile, feet, eyes, and kidney function. If the patient fails to achieve the agreed upon target values, the physician and patient will decide upon changing the therapy.

Typically it takes about 3 months to obtain a stable configuration of target values and monitoring. When a stable configuration has been obtained, the physician will perform a check-up at least every 3 months in which special attention is given to complaints, problems with adaptation of lifestyle, body weight, glucose regulation and blood pressure. Once a year a more thorough check-up is performed in which all aspects of diabetes and its treatment are evaluated.

To ensure that certain steps are taken, caregivers must follow a particular workflow when performing an examination. Starting a new examination triggers a new workflow that enforces that

[10] http://www.portavita.eu/

[11] http://www.dvn.nl

[12] http://www.diabetesfederatie.nl/

```
possible_at(select(classify_feet, recommend(feet_check_up(Frequency)))=true, T):-
   possible_at(diagnosed(diabetic)=true, T),
   possible_at(feet_check_up('Simms', Frequency)=true, T).

% Class 3
possible_at(diabetic_feet_Simms_class(3)=true, T):-
   possible_at(amputated(F)=true, T),
   foot(F).
possible_at(diabetic_feet_Simms_class(3)=true, T):-
   possible_at(ulcer(F)=true, T),
   foot(F).

% Class 2
possible_at(diabetic_feet_Simms_class(2)=true, T):-
   possible_at(neuropathy_condition(loss_of_protective_sensibility)=true, T),
   (
    possible_at(arterial_condition(peripheral_arterial_disease)=true, T);
    possible_at(signs_in_feet(locally_increased_pressure)=true, T)
   ).

% Class 1
possible_at(diabetic_feet_Simms_class(1)=true, T):-
   possible_at(neuropathy_condition(loss_of_protective_sensibility)=true, T).
possible_at(diabetic_feet_Simms_class(1)=true, T):-
   possible_at(arterial_condition(peripheral_arterial_disease)=true, T).

% Class 0
possible_at(diabetic_feet_Simms_class(0)=true, T):-
   \+ possible_at(diabetic_feet_Simms_class(3)=true, T),
   \+ possible_at(diabetic_feet_Simms_class(2)=true, T),
   \+ possible_at(diabetic_feet_Simms_class(1)=true, T).

% Priorities between conclusions
higher_priority(diabetic_feet_Simms_class(X), diabetic_feet_Simms_class(Y)):-
   more_severe(X, Y).

% Higher classes
more_severe(X, Y) :- X > Y.
```

Listing 12: Formulation of diabetic foot based on the Simm's classification.

several steps are taken. When entering a new examination, the user (e.g., healthcare personnel) is required to enter certain data in the examination form. What is required depends on the examination and the entered values. For example, if the user enters that the patient smokes, then he must also enter how much. Fig. 6 shows a part of the foot examination, where the outcomes of several tests can be entered. On the right, previous observation values are showed to clarify whether a condition has changed.

When the user is done entering observations, the examination must be accorded before it is stored in the patient's health record. However, examinations can also be submitted without being accorded. In this case, the examination still has to be accorded before it is stored. For example, an assistant could enter the examination, but the treating physician has to accord it before it is stored in the patient's health record. During the process of entering an examination the user can pause the examination. The examination then appears on the *worklist* of the user. Additionally, there is functionality to plan examinations and to request an examination at another organization or a colleague.

After an examination is saved, it is available to all other users that are involved in the patient's treatment. Various examinations are supported that are particularly relevant to diabetes such as risk inventory, annual check-up, routine check-up, foot examination, interim check-up, self-check, laboratory test, dietary advice, and fundus image screening.

The physician can also register what medication a patient takes. The Anatomical Therapeutic Chemical (ATC) Classification System is used to denote what medication is administered. The ATC is a pharmaceutical coding system that describes drugs by their therapeutic and chemical characteristics. It is controlled by the WHO and

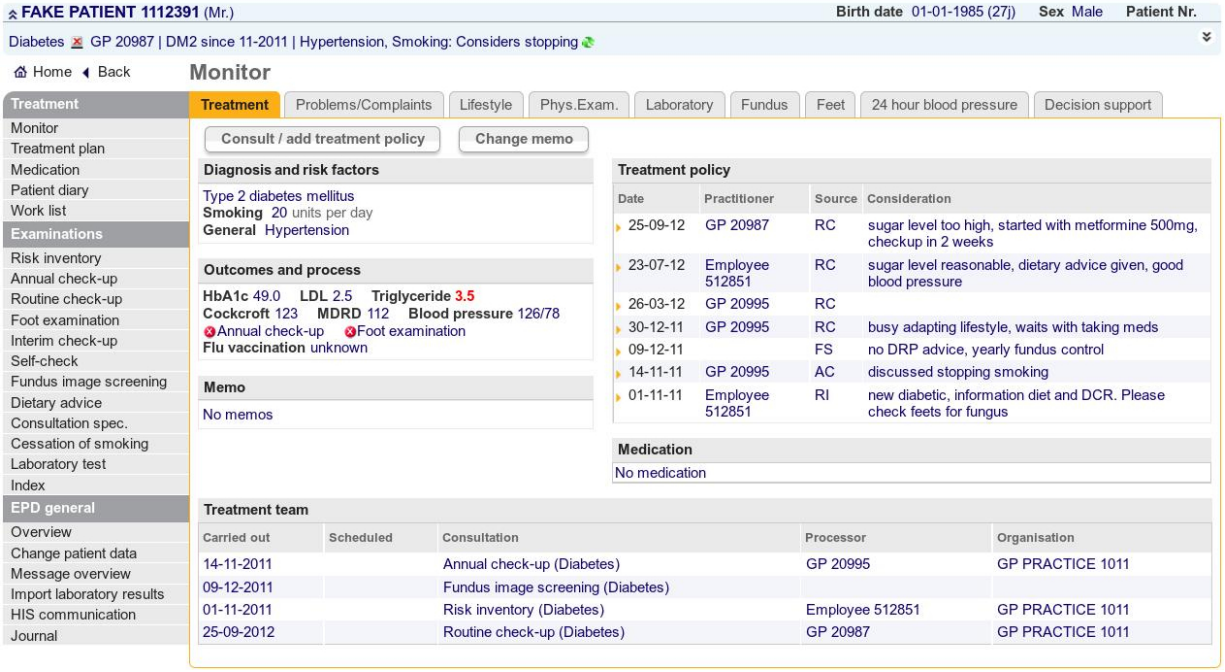


Fig. 8. The monitor gives an overview of the patient's condition.

was first published in 1976. The user has to select what drug, how many doses per day, the quantity per dose (e.g. how many pills), and the start and end date of the medication.

Because different caregivers can enter new information about a patient, it is important to be able to get an up-to-date overview of the patient's condition. The *monitor* is the part of the application that shows the information of a patient that is most relevant to his treatment as can be seen in Fig. 8. There is a section with the *diagnosis and risk factors* of the patient and another named *outcomes and process* that summarizes the most important measurements and whether certain examinations have been performed. In the *memo* section, caregivers can enter messages to each other. The *treatment policy* section provides the dates and remarks of the most important examinations that were performed. The *medication* section shows the medication that the patient takes and finally, the *treatment team* sections shows the caregivers relevant to the patient's treatment. A special part of the monitor is the *decision support* tab (see Fig. 9) in which the user is presented the alerts generated by the COMMODITY$_{12}$ system.

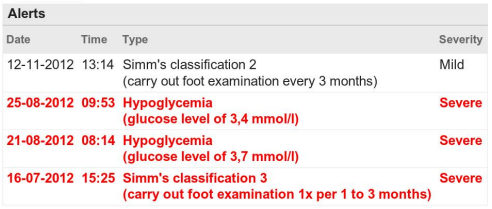


Fig. 9. Alerts concerning the treatment.

6. Related Work

In this section, we review the literature on medical sensors, diabetes management, (agent-based) automated diagnosis and other interoperability solutions for e-health.

6.1. Medical Sensors

Multi-parametric approach to Diabetes monitoring: This research takes into account the need expressed by medical professionals to monitor more than one single parameter in order to be able to better judge the evolution of diabetes 1 or 2 patients. To guarantee sensor reliability and data quality the use of reliable sensors was prioritised due to the need to secure the clinical trial and patient monitoring success. Within the aforemen-

tioned remit this section highlights the most relevant related work to the sensors and wireless connectivity of this project.

Available wireless platforms: Modular wireless sensor platforms represent great potential for laboratory research. The most relevant in this category is the Shimmer platform [17]. Unfortunately Shimmer modules are too large for practical use in a clinical trial, this being caused principally by the choice of relatively high power consumption Bluetooth and ZigBee wireless protocols requiring larger batteries than desired. Accelerometers have been proven to be reliable sources of information for activity monitoring [39] while other approaches involving surface electromyography to detect muscle fatigue have been presented in literature [1] yet the use of electrodes still represents a hurdle for user acceptance. Further commercial products for physiological monitoring exist such as the Zephyr BioHarness 3 [13] and the Equivital EQ02 Life Monitor [14]. These devices use Bluetooth and hence are limited in the duration that they can monitor wirelessly in a continuous fashion. Nevertheless they are worthy of mention as smart duty cycling of such transmission would result in power savings.

Finally, given the need to provide monitoring for long periods of time lower power consumption wireless network protocols were preferred as a first approach over options such as the Shimmer system or others relying on Bluetooth.

Continuous glucose monitors: The field of Continuous Glucose Monitors has received a large amount of attention and focus from medical device companies and several wireless products have recently been announced such as the Dexcom G4 Platinum monitor[15]. This monitor has wireless capabilities but only transmits to a dedicated pocket base station. Dexcom is currently working on the new G5 model that is described as being able to connect to an Android smart phone in a wireless fashion. Because the G5 product has not reached the market yet it could not be included in the presented research at this point in time but will be followed closely for potential interoperability with the smart hub. Further CGMs with wireless capabilities such as the Abbott Freestyle Navigator II [16] present a step forward in wireless continuous glucose monitoring for users but are still limited for interoperability with other systems given their reliance on proprietary data communication schemes. Considering that the pocket base station of the G4 model did not provide a significant benefit over other non-wireless CGMs and that the Medtronic CGM is already available at the DM1 clinic of this project the later sensor was preferred.

6.2. Healthcare Management Systems

The user experience of e-health environments for diabetes has been addressed before [40]. The authors investigate diabetes self-management practices and possible impact factors on future lifestyle choices via a set of experiments. As a result, they discuss opportunities and outline possible directions for the design of e-health applications for diabetes. They argue that pervasive computing, sensor networks and monitoring devices can be combined with machine learning techniques for individuals to keep track of their actions in an unobtrusive way and to provide ground for informed decisions. Furthermore, the authors propose a decision cycle within the diabetes management which can be employed in e-health systems: (i) keep track of performed actions, (ii) monitor blood sugar level, (iii) identify any changes in blood sugar, (iv) attribute the change to particular actions, and (v) modify behaviour based on learned inferences. This work is fundamental for COMMODITY$_{12}$ since on one hand it stresses a patient-centric design, and on the other hand, it provides empirical justification for the development and the deployment of the smart e-health environment presented in this paper.

In [11], the authors provide a conceptual framework for a communication software architecture to realise mobile-device computing and sensor integration. Here, the main objective is to create a transparent and timely communication model that allows different mobile devices to perform different tasks such as patient monitoring and intervention in the health-care process. The model presented integrates its own tailored ontologies for knowl-

[13] http://www.zephyr-technology.com/products/bioharness-3/

[14] http://www.equivital.co.uk/products/tnr

[15] http://www.dexcom.com

[16] https://www.abbott-diabetes-care.de/de-de/produkte/kontinuierliches-glukose-messsystem/freestyle-navigator-ii/

edge generation and pattern-based software engineering. Similar to the COMMODITY$_{12}$ e-health environment, its architecture is layered. The layering is based on a semantic overlay network implemented as a distributed application by means of an object oriented middle-ware for distributed systems. This allows a wide variety of devices to be connected. In contrast to COMMODITY$_{12}$, the authors emphasise the theoretical part of such an ambient intelligence environment: details about the sensors, data-processing and end-user evaluation are omitted.

The SINDI system [41] deploys a logic-based model to reason about different pieces of knowledge for context-aware situation assessment. The focus is on health-care in an assisted-living context for people without a chronic disease. Although the domain is different from COMMODITY$_{12}$, the functional purposes of the two systems are similar: different data sources need to be gathered, integrated and processed by a reasoning engine. Combining these data with a model of medical knowledge, SINDI identifies and prevents risks via declarative feedback policies. In their approach, Answer Set Programming is used instead of the meta-logic framework presented here. Another related application that focuses on the domain of assisted living is presented in [60]. The authors propose an approach for structuring knowledge and reasoning for high-level interpretation of sensor data. They follow a knowledge-based approach that includes rules and ontologies. Instead of using EC for temporal reasoning, they rely on a simplified generalisation of events. Their work bridges the gap between this high-level reasoning and raw sensor data with an abstraction layer that employs numerical reasoning for symbolic conclusions. In contrast to COMMODITY$_{12}$, they employ very basic sensors such as thermometers and sensors that check whether doors are open or closed.

Another important area of research in diabetes management is activity recognition [47]. Here, the authors present a prototypical application, broad to the context of mobile and pervasive computing, where a Hidden Markov Model is fitted to a user's location and activities. Given this model, the authors try to predict the state of activity and the user's location. By using heuristics and tuning parameters of the model, the authors try to enrich the model with the context of a specific user and a specific situation, e.g., going to work or exercising. The long-term objective of this study is to improve support for online learning and context-awareness of individual users and to predict future activities to prevent blood glucose level swings. The work differs from ours in two major points: first, we follow a more top-down approach. We try to start from the medical point of view in terms of treatment and enhance these viewpoints with technical solutions. Second, we employ a more sophisticated sensor system.

6.3. Medical Diagnosis and Agents

From a technical point of view, diagnosis is defined as the process of interpreting the observations that are received from a system domain, and identifying whether there are any faults associated with the system by comparing the expected behaviour of the system with what has been observed [38]. Diagnostic problem solving can be applied to a variety of domains like electronics [57], medical systems [23], and software debugging [6].

Similarly, medical diagnosis is the process of identifying a disease based on the symptoms that are observed on a patient [36]. There are different approaches to medical diagnosis. In fault-based (heuristic) approaches, the idea is to encode the diagnostic reasoning of human experts in a given domain. The real-world system is not modelled. All known faults are modelled instead. Conversely, model-based diagnosis (MBD) starts from a model of the structure (components and their connections), and a set of observations indicating a normal behaviour. A system is then considered faulty if the observed behaviour of the system contradicts the expected (normal) behaviour.

The model-based approach to diagnosis has been started out to identify faults in electronic circuits, for which considerable amount of work has been performed by de Kleer [19]. The idea is to describe the system with a model, and provide a function that accounts for the normal behaviour [38]. This formal theory of diagnosis is called consistency-based diagnosis by Reiter [48]. Consistency-based diagnosis models the normal (expected) behaviour of the system using a logic framework, and it identifies faults through discrepancies between the predicted normal behaviour and the observed abnormal behaviour.

Another computational method to employ diagnosis is to use abduction, in which a model of

the abnormal behaviour is used in terms of cause-effect relations [46,18]. In the idea of abduction, reasoning flows from effects to causes. In a deductive logic program with the rule $P \rightarrow Q$, one can conclude Q whenever the condition P is in the knowledge base. In abduction, however, the aim is to assume that the condition P must be in the knowledge base whenever one observes Q in the environment, giving the rule a semantics of "P has caused Q".

Another alternative for performing model-based diagnosis in the medical domain is to combine the ideas used in deduction and abduction. In a general sense, the medical guidelines are represented by forward rules from symptoms to diseases. If all the necessary symptoms are indeed observed on the patient, then the corresponding disease is drawn as a conclusion, as in classical deduction. However, if some of the symptoms are observed and others are not, the remaining symptoms are assumed to hold, as in abduction. This set of assumed symptoms are then given as advice to the doctor, so that they can be concluded via further relevant tests in order to classify the symptoms into a disease. Enumerating defects and describing these defects via a set of observable findings corresponds to what Lucas calls an abnormality classification diagnosis [37].

In recent years, the model-based diagnosis approach has been applied to multi-agent systems diagnosis [18], with applications in the automotive industry [45], plan diagnosis [50], and coordination failures in agent teams [29,30]. These are in general closed systems, where the agents or components included in the system are fixed during execution. Agent-based approaches are not new for the medical domain. They are used in the literature to manage community health care [9], retrieve medical knowledge [8], decision support systems for monitoring and diagnosing diseases [35], distributed patient scheduling within a hospital [20] and improving the coordination between hospitals for efficient management of organ transplants [3]. There are also agent-based frameworks developed solely for the purpose of improving health care, some of which are ASPIC [22], SAPHIRE [34], K4CARE[17].

6.4. Interoperability Solutions

The epSOS project[18] aims at creating an integration broker for cross border exchange of patient's health records. In epSOS there is no mechanism defined to handle the subscription of new communities, thereby the responsibility to connect different healthcare providers falls into the epSOS operator. In our framework, the use of Mirth Connect solution can be seen as the epSOS broker, with the difference that Mirth Connect can define channels to connect the various entities in the system and it works as an interoperability mediator.

The MediCoordination Healthcare Infrastructure (MHI) [4] aims at improving the accessing and sharing of important medical data between medical actors. MHI's architecture consists of a registry and two clients, one for submitting documents and one for receiving them. An XDS-based server is used for the repository and the registry. The IHE XDS Integration Profile describes an infrastructure based on standards (ebXML), for managing the information exchange of sensitive medical data. The MHI prototype does not implement notifications [4]. General practitioners have to manually query the registry and client-server communications are channelled through a SOA-based service. With respect to MHI, our infrastructure hides to the actors the complexity of using the infrastructure by defining a publish and subscribe mechanism between the involved components (Smart Hub, GOLEM, Mirth Connect, Web Services).

As reported in [21], ARTEMIS is a project that provides interoperability for healthcare by semantically enriching Web services. Such semantically enriched web services are then connected by means of the JXTA P2P infrastructure. A JXTA super peer represents a community where different health actors can interact. The ontologies in the medical domain are mapped directly to different ontologies related to the heterogeneous IT systems making use of ARTEMIS. A set of mediator components is then used to consent heterogeneous services to communicate. With respect to ARTEMIS, we do not make use of semantic representations, as our complex event processing components rather focuses on detecting patterns in

[17] `http://www.k4care.net`

[18] `http://www.epsos.eu`

the physiological events produced by the patient. Furthermore, the problem of representing health concepts is already addressed by standards like SNOMED CT[19], which within COMMODITY$_{12}$ are used in CCD messages that are then interpreted by the web services.

With respect to ARTEMIS, which only focuses on the communication amongst the components in the infrastructure, our agents are dedicated to perform reasoning on the event patterns associated with the condition of the patient. As a matter of fact, in COMMODITY$_{12}$ we aimed at creating a platform that could support any kind of chronic illness, by separating the concerns: COMMODITY$_{12}$ implements a distributed event-based system where the agents are subscribers of the events produced by the patient. The physiological values of the patient are represented using standards such as CCD, which allows us to hide the complexity of interfacing the monitoring sensors with the smart hub solution and to translate the messages to the language understood at the agent level (Prolog terms).

The Provenance project [31] defines a multi-agent system that records the patient's complete health care history (located in different communities). Each community creates and processes the documentation using p-assertions that are stored in a Provenance store. The p-assertions are assertions triggered by the documentation of a patient's health treatment process and indicate information such as which is the medical doctor who generated a record, what were the basis for the given treatment, and when the record was created. Rather than using p-assertion, we use CCD to represent our messages, with the addition of SNOMED CT and LOINC codes. Furthermore, we do not only have doctors producing information about patients, but also the patients themselves, by using the smart hub, can produce CCD messages containing medical information.

Similarly to COMMODITY$_{12}$, Provenance performs event processing on the events produced by the patients and the doctors. Differently from COMMODITY$_{12}$, Provenance does not define complex event pattern components, such as our $\mathcal{LAMA}$ agents, to handle complex event patterns produced by the patients. Also, the reasoning performed in Provenance focuses on defining triggering rules on the current health records of the patient. In COMMODITY$_{12}$ we consider temporal patterns both on the long and short term: the decision taken by the agent is associated with the physical examinations of the patient at the doctor side, on the lab tests evolution, and on the physiological values collected with the smart hub component.

Triple space computing (TSC) applied to healthcare [43] uses tuple spaces to foster the exchange of information and proposes the use of semantic web technology to represent the data about the patients. Their solution associates RDF tuples to concepts defined in HL7 or SNOMED. This solution differs from ours as we are not concerned with translating HL7 concepts into a semantic web language. Here, we are interested in defining complex event processing on the multi-parametric physiological values of the patient, rather then semantic reasoning amongst concepts expressed in RDF. From the perspective of the computation, also TSC considers the problem of publication and retrieval of health information, but it does not describe the notification and dispatching of the events happening in the distributed systems, while here we use GOLEM and Mirth Connect to handle the publication and subscription of events.

In particular, our approach has a deeper granularity than the one proposed in TSC. As a matter of fact, COMMODITY$_{12}$ focuses on a multi-layered approach where our agents represent the subscribers to the events produced by the patient. In this way, agents are left with the task of reasoning about the data produced by the patient, rather than having to deal with infrastructure problems such as in TSC and in other approaches.

In [58], Tentori et al. present a pervasive healthcare infrastructure that takes into consideration the problem of privacy of the patients. The platform agent presents to patients, when they interact with the platform, several privacy choices when handling their data. The interaction between the patient and the intelligent agents is handled by means of a negotiation protocol. With respect to [58], COMMODITY$_{12}$ utilises the Portavita platform to deal with privacy concerns of the user. Portavita represents the web interface of the patient. In particular in COMMODITY$_{12}$ the final owner of the data is always considered to be the

[19] http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html

patient, who is the only one that decides which party can have access to the data. In principle, the patient could use COMMODITY$_{12}$ as a diabetic diary without allowing doctors to observe his/her physiological values.

Finally, in [59], Tentori et al. present an activity recognition framework based on the SALSA agent library [49]. The framework proposed in [59] is the most similar to COMMODITY$_{12}$ in terms of architecture as it implements a distributed event-based system where the agents perform reasoning on the activity performed by the patients. Differently from COMMODITY$_{12}$, the framework presented in [59] utilizes Hidden Markov Models (HMM) to classify the activity of the patient. For the moment, in COMMODITY$_{12}$ we put most of our effort on translating guidelines to monitoring rules, so to be able to apply the guideline on the physiological value of the patient. Machine learning approaches towards patient diagnosis are currently planned for the COMMODITY$_{12}$ project. However, the focus will be on methods that are logic-based or that can use data to provide an explanation on the current situation of the user by means of causality models, such as in [53]. As this is of increasing importance to medical doctors experts in diabetes, who need to understand how the relationship amongst the physiological values of the patient evolves in time so that to apply a better treatment to patients.

7. Conclusions

We have presented a Personal Health System (PHS), called COMMODITY$_{12}$, that assists in the provision of continuous and personalised health services to diabetic patients, with the aim of empowering their lifestyle regardless of their location. COMMODITY$_{12}$ consists of ambient, wearable and portable devices, which acquire, monitor and communicate physiological parameters and other health-related context of an individual, such as physical activity and vital body signals. The data are interpreted by personal agents that use expert biomedical knowledge to derive important insights about the individual's health status, which are then presented in the form of active feedback to the patient directly from the device, or via health professionals who assist in diagnosis, treatment and life management.

Our work has focused on the design of the PHS in terms of its main components, their integration and deployment to address major problems of interest to both diabetic patients and doctors that treat diabetes. In this context, our contribution is as follows.

- We have presented a generic system architecture for a PHS specialised on continuous monitoring of diabetes patients. The practical significance of the architecture is that it integrates sensors, intelligent agents, databases and users within a single system. The approach taken differentiates between domain-specific and generic requirements for such a PHS and develops the components of the framework so that they can be reused in a different domain by integrating the necessary sensors and medical knowledge accordingly.
- We have identified a set of commercially available sensors that aid the monitoring of diabetes patients and have discussed how these sensors can be organised to provide input to the overall system.
- We have specified a model and architecture, called $\mathcal{LAMA}$, to develop software agents that can reason about the medical domain of diabetes to support patients and medical personnel. $\mathcal{LAMA}$ agents are endowed with useful reasoning capabilities and a knowledge representation framework that can support a range of requirements, from monitoring to diagnosis.
- We have discussed how to deploy $\mathcal{LAMA}$ agents through the use of the GOLEM platform, thus allowing the distribution of PHS components over a complex computer network.
- We have shown how to integrate different components via a mediator that translates information between agents, sensors, medical databases, web services and user interfaces, thus making the PHS robust and extensible.
- We have grounded the ideas, models, techniques and technologies through the development of two case study scenarios on diabetes management.
- We have compared and evaluated our approach with relevant work in the literature for all the components and technologies used in the proposed PHS.

As part of our future work we plan to include the patient's profile more actively in the treatment process. The profile of a patient contains clues on the lifestyle and the surroundings of the patient as well as her medical history. It is very important that patients get the suitable treatment plan tailored to their needs and lifestyle. The profile information can be used to detect exception cases that distinguishes a patient from one with a "normal" profile. One such exception is about the working hours of a patient. For example, if a patient works late from midnight to morning, then the doctor may have to deviate from a normal treatment plan.

Another important future goal is to introduce uncertainty to the reasoning process of agents by specifying a data-driven capability that takes advantage of the data collected live via the BAN and available in the current system database containing health-records of 20,000 diabetic patients. We have already made some progress with analysing big-data of this kind with a multi-dimensional causal discovery method that we have introduced in [53], but additional work is required in this important area.

We also plan to perform a monitored end-user evaluation through clinical trials carried out by the doctors of our consortium. These clinical trials will be conducted with peer-reviewed protocols that are approved by ethical committees, thus guaranteeing high scientific standards. Here we will examine a considerable number of patients in two European countries, divided in treatment and control groups. For evaluating the system, we predefine the following outcome measures. First, we would like to assess objective improvement of diabetes management. This can be measured with the percentage of complications associated to diabetes type 1 and 2 correctly identified by the system. Then, we would like to evaluate the number of complications associated with cardiovascular co-morbidities that have been correctly identified by the system. We would also like to see how the patients respond to the different treatments and assess if the platform can provide the correct treatment adjustment.

In the longer term, we want to determine the effects of the system on the quality of life of participants via questionnaires, analyse how patients deal with the use of sensors and report our findings to the e-health community. We also want to study the effort of moving our approach from one domain to the other, from sensors and databases to programming agents with different domain specific knowledge.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in a previous version of this article. This work was partially supported by the EU FP7 Project COMMODITY$_{12}$ (`www.commodity12.eu`).

References

[1] M.R. Al-Mulla, F. Sepulveda, and M. Colley. An autonomous wearable system for predicting and detecting localised muscle fatigue. *Sensors*, 11(2):1542–1557, 2011.

[2] K.G.M.M. Alberti and P.Z. Zimmet. Definition, diagnosis and classification of diabetes mellitus and its complications. Part 1: diagnosis and classification of diabetes mellitus. Provisional report of a WHO Consultation. *Diabet. Med.*, 15(7):539–553, 1998.

[3] A. Aldea, B. Lopez, A. Moreno, D. Riano, and A. Valls. A multi-agent system for organ transplant co-ordination. In Silvana Quaglini, Pedro Barahona, and Steen Andreassen, editors, *Artificial Intelligence in Medicine*, volume 2101 of *Lecture Notes in Computer Science*, pages 413–416. Springer Berlin, Heidelberg, 2001.

[4] B. Alves, H. Muller, M. Schumacher, D. Godel, and O. Abu Khaled. Interoperability prototype between hospitals and general practitioners in Switzerland. *Stud Health Technol Inform*, 160:366–370, 2010.

[5] J. Apelqvist, J. Bakker, W. H. van Houtum, and N. C. Schaper. Practical guidelines on the management and prevention of the diabetic foot. *Diabetes Metab Res Rev*, 24(Suppl 1):181–187, 2008.

[6] L. Ardissono, L. Console, G. Goy, G. Petrone, G. Picardi, M. Segnan, and D. T. Dupré. Enhancing web services with diagnostic capabilities. In *Proceedings of the 2005 IEEE International Conference on Web Services (ICWS 2005)*, pages 182–191. IEEE Computer Society, 2005.

[7] A. Artikis, M. Sergot, and J. Pitt. Specifying norm-governed computational societies. *ACM Trans. Comput. Logic*, 10(1):1–42, 2009.

[8] O. Baujard, V. Baujard, S. Aurel, C. Boyer, and R.D. Appel. MARVIN, multi-agent softbot to retrieve multilingual medical information on the web. *Informatics for Health and Social Care*, 23(3):187–191, 1998.

[9] M. Beer, R. Hill, W. Huang, and A. Sixsmith. An agent-based architecture for managing the provision of community care - the INCA (intelligent community alarm) experience. *AI Commun.*, 16(3):179–192, 2003.

[10] T. Benson. *Principles of Health Interoperability HL7 and SNOMED*. Springer, 2009.

[11] J. Bravo, V. Villarreal, R. Hervas, and G. Urzaiz. Using a communication model to collect measurement data through mobile devices. *Sensors*, 12(7):9253–9272, 2012.

[12] S. Bromuri, M. Schumacher, K. Stathis, and J. Ruiz. Monitoring gestational diabetes mellitus with cognitive agents and agent environments. In *Proceedings of the 2011th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2011)*, 2011.

[13] S. Bromuri and K. Stathis. Situating cognitive agents in GOLEM. In *Engineering Environment-Mediated Multi-Agent Systems, International Workshop*, pages 115–134, 2007.

[14] S. Bromuri and K. Stathis. Distributed agent environments in the ambient event calculus. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 12:1–12:12, New York, NY, USA, 2009. ACM.

[15] S. Bromuri, V. Urovi, and K. Stathis. Game-based e-retailing in golem agent environments. *Pervasive Mob. Comput.*, 5(5):623–638, 2009.

[16] S. Bromuri, V. Urovi, and K. Stathis. iCampus: A Connected Campus in the Ambient Event Calculus. *International Journal of Ambient Computing and Intelligence*, 2(1):59–65, 2010.

[17] A. Burns, B.R. Greene, M.J. McGrath, T.J. O'Shea, B. Kuris, S.M. Ayer, F. Stroiescu, and V. Cionca. Shimmer; a wireless sensor platform for noninvasive biomedical research. *Sensors Journal, IEEE*, 10(9):1527 –1534, 2010.

[18] L. Console, D.T. Dupré, and P. Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[19] J. De Kleer, A.K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[20] K. Decker and J. Li. Coordinated hospital patient scheduling. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 104–111. IEEE, 1998.

[21] A. Dogac, G.B. Laleci, S. Kirbas, Y. Kabak, S.S. Sinir, A. Yildiz, and Y. Gurcan. Artemis: deploying semantically enriched web services in the healthcare domain. *Inf. Syst.*, 31:321–339, 2006.

[22] J. Fox, D. Glasspool, and S. Modgil. A canonical agent model for healthcare applications. *IEEE Intelligent Systems*, 21(6):21–28, 2006.

[23] J. Gamper and W. Nejdl. Abstract temporal diagnosis in medical domains. *Artificial Intelligence in Medicine*, 10:209–234, 1997.

[24] A. Hinchley. *Understanding Version 3: A Primer on the HL7 Version 3 Communication Standard*. Understanding HL7. Alexander Mönch Publishing, 2005.

[25] K. V. Hindriks and M. B. van Riemsdijk. Satisfying maintenance goals. In *6th International Workshop on Declarative Agent Languages and Technologies (DALT)*, pages 86–103, 2008.

[26] A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Declarative agent control. *Computational Logic in Multi-Agent Systems*, pages 148–148, 2005.

[27] A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *J. Log. Comput.*, 2(6):719–770, 1992.

[28] A.C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Computational logic foundations of KGP agents. *J. Artif. Intell. Res. (JAIR)*, 33:285–348, 2008.

[29] M. Kalech and G. A. Kaminka. Diagnosing a team of agents: scaling-up. In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 249–255, 2005.

[30] G. A. Kaminka. Detecting disagreements in large-scale multi-agent teams. *Autonomous Agents and Multi-Agent Systems*, 18(3):501–525, 2009.

[31] T. Kifor, L.Z. Varga, J. Vazquez-Salceda, S. Alvarez, S. Willmott, S. Miles, and L. Moreau. Provenance in agent-mediated healthcare systems. *Intelligent Systems, IEEE*, 21(6):38 –46, 2006.

[32] R.A. Kowalski. Logic without model theory. *What is a logical system*, pages 35–71, 1993.

[33] R.A. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.

[34] G. B. Laleci, A. Dogac, M. Olduz, I. Tasyurt, M. Yuksel, and A. Okcan. Saphire: A multi-agent system for remote healthcare monitoring through computerized clinical guidelines. In *Agent Technology and e-Health*, Whitestein Series in Software Agent Technologies and Autonomic Computing, pages 25–44. 2008.

[35] J.E. Larssan and B. Hayes-Roth. Guardian: intelligent autonomous agent for medical monitoring and diagnosis. *Intelligent Systems and their Applications, IEEE*, 13(1):58 –64, 1998.

[36] R.S. Ledley and L.B. Lusted. Reasoning foundations of medical diagnosis. *Science*, 130(3366):9–21, 1959.

[37] P. Lucas. Symbolic diagnosis and its formalisation. *The Knowledge Engineering Review*, 12(2):109–146, 1997.

[38] P.J.F. Lucas. Analysis of notions of diagnosis. *Artificial Intelligence*, 105(1):295–343, 1998.

[39] M. Lustrek, B. Cvetkovic, and S. Kozina. Energy expenditure estimation with wearable accelerometers. In *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pages 5–8, 2012.

[40] L. Mamykina, E.D. Mynatt, and D.R. Kaufman. Investigating health management practices of individuals with diabetes. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 927–936. ACM, 2006.

[41] A. Mileo, D. Merico, and R. Bisiani. Support for context-aware monitoring in home healthcare. *Journal of Ambient Intelligence and Smart Environments*, 2(1):49–66, 2010.

[42] M. Morge, J. McGinnis, S. Bromuri, P. Mancarella, K. Stathis, and F. Toni. Argumentative agents for service-oriented computing. In Maria Ganzha and Lakhmi C. Jain, editors, *Multiagent Systems and Applications*, volume 45 of *Intelligent Systems Reference Library*, pages 217–255. Springer Berlin Heidelberg, 2013.

[43] L.J.B. Nixon, D. Cerizza, E. Della Valle, E.P.B. Simperl, and R. Krummenacher. Enabling Collaborative eHealth through Triplespace Computing. In *WETICE*, pages 80–85, 2007.

[44] Nordic Semiconductor. nRF24LE1 - ultra-low power wireless system on-chip solution - product specification v1.6, August 2010.

[45] C. Picardi, R. Bray, F. Cascio, L. Console, P. Dague, D. Millet, B. Rehfus, P. Struss, and C. Vallée. IDD: Integrating diagnosis in the design of automotive systems. In *ECAI 2002: 15th European Conference on Artificial Intelligence*, pages 628–632. IOS Press, 2002.

[46] D. Poole. Normality and faults in logic-based diagnosis. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI) - Volume 2*, pages 1304–1310, 1989.

[47] D. Preuveneers and Y. Berbers. Mobile phones assisting with health self-care: a diabetes case study. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 177–186. ACM, 2008.

[48] R. Reiter. A theory of diagnosis from first principles. *Artificial intelligence*, 32(1):57–95, 1987.

[49] M.D. Rodriguez and J. Favela. Evaluation of an Agent Framework for the Development of Ambient Computing. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, volume 5333 of *Lecture Notes in Computer Science*, page 374–383. Springer Berlin Heidelberg, 2008.

[50] N. Roos, A. ten Teije, and C. Witteveen. A protocol for multi-agent diagnosis with spatially distributed knowledge. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 655–661, 2003.

[51] M.M. Safford, L. Russell, D-C. Suh, S. Roman, and L. Pogach. How much time do patients with diabetes spend on self-care? *Journal of the American Board of Family Medicine*, 18(4):262–270, 2005.

[52] G. Schadow, C.N. Mead, and D.M. Walker. The HL7 reference information model under scrutiny. *Stud Health Technol Inform*, 124:151–156, 2006.

[53] U. Schaechtle, K. Stathis, and S. Bromuri. Multidimensional causal discovery. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (To appear)*. AAAI Press, 2013.

[54] M. Shanahan. The event calculus explained. In *Artificial Intelligence Today*, pages 409–430. 1999.

[55] D.S. Sims Jr., P.R. Cavanagh, and J.S. Ulbrecht. Risk factors in the diabetic foot: Recognition and management. *Phys Ther*, 68(12):1887–1902, 1988.

[56] K. Stathis, S. Kafetzoglou, S. Pappavasiliou, and S. Bromuri. Sensor Network Grids: Agent Environment combined with QoS in Wireless Sensor Networks. In *Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems (ICAS'07)*, pages 47–, Washington, DC, USA, 2007. IEEE Computer Society.

[57] A. Subbiah and D. M. Blough. Distributed diagnosis in dynamic fault environments. *IEEE Transactions on Parallel and Distributed Systems*, 15:453–467, 2004.

[58] M. Tentori, J. Favela, and M.D. Rodriguez. Privacy-Aware Autonomous Agents for Pervasive Healthcare. *Intelligent Systems, IEEE*, 21(6):55–62, 2006.

[59] M. Tentori, M. Rodriguez, and J. Favela. An Agent-Based Middleware for the Design of Activity-Aware Applications. *IEEE Intelligent Systems*, 26(3):15–23, 2011.

[60] H.J. ter Horst and A. Sinitsyn. Structuring reasoning for interpretation of sensor data in home-based health and well-being monitoring applications. *Journal of Ambient Intelligence and Smart Environments*, 4(5):461–476, 2012.

[61] Texas Instruments. SimpliciTI Compliant Protocol Stack.

[62] Texas Instruments. MSP430x5xx and MSP430x6xx Family - User's Guide, August 2012.

[63] F. Toni, M. Grammatikou, S. Kafetzoglou, L. Lymberopoulos, S. Papavassileiou, D. Gaertner, M. Morge, S. Bromuri, J. McGinnis, K. Stathis, V. Curcin, M. Ghanem, and L. Guo. The argugrid platform: An overview. In Jórn Altmann, Dirk Neumann, and Thomas Fahringer, editors, *Grid Economics and Business Models*, volume 5206 of *Lecture Notes in Computer Science*, pages 217–225. Springer Berlin Heidelberg, 2008.

[64] W. van Houtum. Personal communication, Nov 2012.

[65] M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: a unifying framework. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 713–720, 2008.

[66] Nederlandsche Internisten Vereeniging. *Richtlijn Diabetische Voet*. Van Zuiden Communications B.V., 2007.