Journal of Ambient Intelligence and Smart Environments 2 (2010) 49–66
DOI 10.3233/AIS-2010-0052
IOS Press

Support for context-aware monitoring in home healthcare

Alessandra Mileo, Davide Merico * and Roberto Bisiani
Department of Informatics, Systems and Communication, NOMADIS Research Lab., University of Milan-Bicocca, Italy, E-mail: {alessandra.mileo,davide.merico,roberto.bisiani}@nomadis.unimib.it

Abstract. This paper tackles the problem of supporting independent living and well-being for people that live in their homes and have no critical chronic condition. The paper assumes the presence of a monitoring system equipped with a pervasive sensor network and a non-monotonic reasoning engine. The rich set of sensors that can be used for monitoring in home environments and their sheer number make it quite complex to provide a correct interpretation of collected data for a particular patient. For this reason, we introduce a logic-based context model for situation assessment combined with high level declarative feedback policy specification, and we use logic programming techniques to reason about different pieces of knowledge for prevention.

Keywords: Independent Living, non-monotonic reasoning, knowledge representation, wireless sensor networks, situation assessment

1. Introduction

Over the last fifteen years, numerous efforts have been made to create IT-based support systems for the elderly. The main objective of these systems was to help elderly people live a safe life while keeping their independence as long as possible. There are many different ways in which such systems have offered support for independent life: for example tele-monitoring chronic pathologies, recognizing activities-of-daily-living and supporting their correct execution, reminding the execution of important activities, jogging memory with exercises, helping movements with robotic assistants, and so on. All these different technological supports should cooperate for one main purpose: prolonging Healthy Life Years. A support system should strive for people to live longer but not in a state in which chronic conditions substantially cripple the quality of life and the capability of living a productive life. Therefore, we think one should stress cognitive support as a way of improving all sides of life from the use of memory to the capability of avoiding falls.

An Independent-Living System (ILS) should be able to (i) gather information about the world through sensors, (ii) translate sensor data to map them into a consistent assessment of the real situation, (iii) reason about available knowledge to support the patient's well being, (iv) perform actions and give feedback to the patient according to the results of the reasoning process, (v) capture reactions to feedback in order to adapt the behavior of the system.

We believe that one of the most useful goals should be to prevent situations that can cause drastic changes for the worse of the quality of living, such as falls and substantial weight loss. Rather than supporting activities [12,13] and observing behaviors [15,39], this paper considers a complementary view of artificial intelligence applied to home healthcare. In our view, expressive knowledge representation and reasoning techniques should be used to (i) understand health evolution identifying risky states in a context-aware fashion and (ii) give clinical, behavioral or environment-related feedback. This can be done by applying automated reasoning to a combination of different pieces of knowledge (common-sense, medical, context-dependent), rather than dealing with predefined plans and goals to be achieved.

[*] Corresponding author.

1876-1364/10/$27.50 © 2010 – IOS Press and the authors. All rights reserved

With this perspective, the capability of identifying meaningful information about the context in which the user lives is a critical issue for home healthcare systems. The use of a reasoning component that does not only rely on static user-specific needs, but that continuously analyzes the evolving state of the patient and of the environment, simplifies the situation assessment process. The aggregation and the interpretation of different kinds of information from heterogeneous sources (such as light, position, movement, localization, load cells) enhances reliability and accuracy of context interpretation because considering heterogeneous sources of information helps in compensating errors and incompleteness of data.

In order to address these concerns, we have designed and developed a system (called SINDI [2]) that has the following capabilities:

- gathering data about the user and his or her environment through a Wireless Sensor Network (WSN);
- reasoning about collected data for context interpretation and situation assessment using commonsense, constraints and qualitative preferences;
- combining different data sources to interpret the evolution of the patient's health state providing absolute and differential evaluations of indicators of well-being as well as of more general human functioning;
- predicting changes into risky states according to a graph-based computational model of medical knowledge and the clinical profile of the person monitored;
- identifying risky situations and providing feedback for prevention through the enforcement of declarative feedback policies.

The first and the second aspects are strictly related to context interpretation based on the model of context described in Section 2. The logical framework used for Knowledge Representation and Reasoning is introduced in Section 3, while the reasoning steps aimed at context interpretation are illustrated in Section 4; the following aspects refer to reasoning about well-being according to the graph-based structure of knowledge about elderly care that we use in SINDI; in Section 5 we introduce the reasoning processes supporting prediction and prevention, and we illustrate in more details the way feedback outputs for prevention are modelled and provided to the patient according to results of prediction. We use the prediction and prevention of falls as an example in Section 6 because it is a problem where both cognitive and physical impairment come to play.

Other important requirements like: (i) technological and medical soundness, (ii) adaptivity, (iii) unobtrusiveness, (iv) user-friendliness, (v) reactivity and (vi) affordability have also been investigated in the specification of our system, but in this paper we focus on our context-aware knowledge representation and reasoning model supporting user-centered prediction and prevention in home healthcare.

2. The context model

A well designed model is crucial for any context-aware system. In the literature there is a rich variety of context models discussed and proposed for different purposes [1,33].

Context-dependent data can arise from different sources; for example data may be gathered by sensors or collected from several knowledge-bases. The incompleteness and heterogeneous nature of such data stress the need for expressive reasoning techniques in order to implement effective, context-dependent reasoning.

While most of the implemented context models are domain-dependent and do not support powerful inference, some logic-based models fail to provide a representation of context-dependent data that is both expressive and with good computational properties [37].

The model we describe in this paper aims at being generic and computationally rich at the same time. Although we do not provide a comprehensive theory of context in this work, this aspect deserves attention and it is under investigation. Other requirements we take into account are:

- *readability* of the problem specification,
- *flexibility* with respect to using heterogeneous sources of knowledge,
- *modularity* in the specification of the problem that describes properties of a desired solution, and
- *expressivity* of the modeling language and computational efficiency of the inference engines.

To fulfill these requirements, we utilize a declarative high-level description of home environments in terms of rooms, areas, objects, properties, relations and observations. The space is represented as a grid where each cell is identified by coordinates X, Y and has some properties (being in a room, being a wall, being a passage, etc.). In addition to the description of the model, a limited set of consistency constraints can be specified to make sure that observations and context interpretation are coherent with the person's profile (e.g. walking speed) and the environment (e.g. walls and obstacles). The resulting context specification is then mapped into a set of logic predicates in the Answer Set Programming (ASP) framework (see Section 3 for preliminary notions), so that reasoning under uncertainty is possible and incompleteness of data can be taken into account.

The physical description of the environment is available in terms of coordinates in a bi-dimensional grid. Properties used in the model take into account generic spatial properties rather than describing geometric spatial relations between objects. This results in greater generality because we do not need a complete physical description of the environment. Generic spatial properties can be used in the inference process to better evaluate the feasibility of a solution. As an example, the system can infer that the person is in a given area A_j of a room R_i on the basis of information provided by motion sensors only. However, coordinates evaluation could return a different localization result, thus the system deduce that the moving object could be a cat or something else.

Occasional lack of specificity w.r.t. the spatial relations is compensated by the inference process: reasoning about relations and attribute values may help inferring new information as detailed in Section 4.

In addition, while data gathered by the sensors are processed and aggregated according to specific algorithms for feature analysis, the information available at upper levels is filtered by the abstraction. This enables us to represent meaningful information as properties of objects, rooms or areas, keeping the model independent from sensors' characteristics and positioning.

Our modelling approach is similar to what we would obtain by using an ontology [26], with the difference that the ASP reasoning enhances readability, expressivity and computational efficiency of the model. We are aware of the fact that research efforts are converging toward the combination of non-monotonic reasoning and ontology-based knowledge representation, but available implementations are still domain dependent and formal issues need to be further explored. For this reason we decided to encode our contextual information directly into first order logic under the ASP framework. The domain description and inference rules can be easily combined with domain knowledge extracted from an (existing or new) ontology if needed.

Previous investigation of context models has indicated that there are certain entities in a context that, in practice, are more important than others. These are location, identity, activity and time [29,31]. In fact, in the context of home monitoring, the more intuitively relevant aspects of a context are: *where* you are, *who* you are (clinical profile), *which* resources you are using, *what* you are doing and *when*.

In order to represent this information in our model we identify the following entities:

- *Person* entity to identify the person, her clinical profile and her movements;
- *Room* entity to identify rooms in the environment;
- *Area* entity to identify disjoint areas of interest in a room;
- *Object* entity to identify objects or resources the person can interact with.

We also define a small subset of generic spatial relations among entities, summarized in Table 1[1]. All the other pieces of information are available at a higher level of detail and can be indexed as attributes of the context entities. Attributes value may come from (i) external knowledge (such as observed values for clinical attributes of the Person entity), (ii) opportunely aggregated sensor data (such as attributes related to the environment) or (iii) results of the inference process (inferred values for attributes of the Person entity, when observed values are not available). Values of both attributes and spatial relations (except the inclusion of an area in a room which is static) are dynamic and need to be associated to a time T. In this way, the reasoning system can take into account their evolution during context interpretation. In such a dynamic domain it is necessary to conceive a model that is able to support commonsense reasoning, reasoning about actions and qualitative preferences.

Attributes associated with entities included in our model and their values are detailed in Tables 2, 3 and 4.

Table 1

Generic spatial relations among entities

Relation Name	Object	Reference Object	Relation Type
person_in	Person	{Room, Area}	generic directional relation
in	Area	Room	generic directional relation
	Object	{Room, Area}	
near	Object	Area	generic distance relation

[1] Note that the spatial inclusion of areas $A_1, \ldots, A_n$ in a room R is such that $\bigcup_{i=1}^{n} A_i \subset R$.

Table 2

Attributes representing information about the Person entity

Signature	Domain Values	Description
<Func,Val>	Func={gait, balance, vision, cognition, bmi, sleep} Val ={absent, mild, moderate, severe}	Functional disability

<Adl,Val>	Adl={mobility, dress, eat}
	Val={ok, needy, dependent}
		ADL dependency

<Risk, Val>	Risk={fall, depression, fragility} Val ={absent, mild, moderate, severe}	Risk assessment
<Test, Val>	Test={amtest, minimental, nutritionTest, gds, visual, . . . } Val={ok, mild, moderate, severe}	Test results
<Drug, Val>	Drug={benzodiazepine, antidepressant, diuretic, ssri, . . . } Val ={yes, no}	Medications
<Disease, Val>	Disease={visual_impairment, arthrosis, depression, . . . } Val={ok, mild, moderate, severe}	Diseases
<wgt, Val>	Val={1..300}	Weight in kilograms
<motion,Val,P[1]>	Val={walk, still, zeroSignal[2]}, P={0..100}	Motion activity
<posture,Val,P[1]>	Val={sit, lay, stand, zeroSignal[2]}, P={0..100}	Posture of the person
<orientation,Val,P[1]>	Val={turn, straight, zeroSignal[2]}, P={0..100}	Direction of motion

[1] Value "zeroSignal" is related to the fact that no signal is received from sensors detecting movement.

[2] Parameter "P" represents data reliability, and it is computed by the algorithms used for feature extraction.

Table 3

Attributes representing information about the Room and Area entities

Attribute Name	Domain Values	Description
ambientLight	{dark, . . . , bright}	brightness of the environment
ambientLightType	{natural, artificial}	nature of the light
ambientHumidity	{dry, medium, wet, superWet}	humidity level
ambientTemperature	{cold, chilly, warm, hot, burning}	temperature
ambientSound	{mute, mild, medium, noisy}	noise level
presence	{yes, no}	presence of a moving entity
smoke	{yes, no}	presence of smoke

Table 4

Attributes representing information about the Object entity

Signature	Domain Values	Description
objectLight	{dark, . . . , bright}	light produced by the object
objectLightType	{natural, artificial}	nature of the light
objectSound	{noSound, regularSound, loudSound}	meaning depends on object
switch	{open, closed}	state of doors and windows objects
state	{on, off}	state of on/off devices
filteredLoad	{0..300}	weight measurement from load-cells
loadVolatility	{stable, mildlyUnstable, veryUnstable}	volatility of the load measurement

Most of the values for attributes associated to rooms, areas and objects are the result of a process that maps numerical sensor data into meaningful thresholds. The thresholds are derived both from objective considerations, e.g. a given temperature might be too hot for the human body to survive, and from patient-profile and environment related considerations, e.g. a southern Italian and a British person might have a very different idea of what is comfortably hot or cold.

As an example, if we want to model the environment illustrated in Fig. 1 at time T, the partial description we obtain by instantiating our first order specification

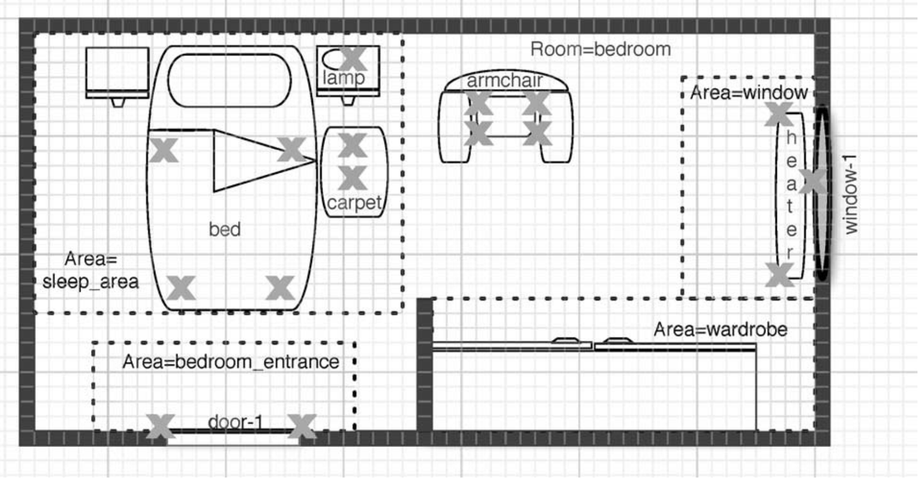

Fig. 1. Example: modelling a bedroom.

in terms of propositional predicates, would be similar to what follows:

```
obj(door). obj(lamp). obj(carpet). obj(heater).
obj(armchair). obj(window). obj(bed).
area(sleep_area). area(wardrobe). area(entrance).
room(bedroom).

in(sleep_area,bedroom,1).
in(wardrobe,bedroom,1).
in(entrance,bedroom,1).

in(door,bedroom,1). in(heater,bedroom,1).
in(lamp,sleep_area,1). in(carpet,sleep_area,1).
in(armchair,bedroom,1). in(window,bedroom,1).
in(bed,sleep_area,1).

objectLight(lamp,dark,1).
switch(door_1,open,1). switch(window_1,closed,1).
filteredLoad(armchair,0,1).
filteredLoad(carpet,0,1).
filteredLoad(bed,0,1).

...
```

Provided this description, commonsense rules, non-monotonicity and preferences can be used for context interpretation. These aspects will be illustrated in Section 4.2. We now provide some preliminary notions that are needed in order to understand the strength of the ASP formalism and the way we use it.

3. The logical framework: Preliminary notions

The declarative logical framework we use for Knowledge Representation and Reasoning in SINDI is that of Answer Set Programming (ASP), based on the *stable model* semantics for Logic Programs proposed by Gelfond and Lifschitz [10].

Compared to pure statistical approaches, logic inference based on ASP is highly expressive and computationally more performant because it can deal with first-order representations, which are much richer than the propositional ones characterizing probabilistic inference. Furthermore, ASP can deal with incomplete information and commonsense reasoning using defaults. Cardinality and weight constraints together with optimization techniques are also interesting features for our application, in that they can be used to model different degrees of uncertainty [5,9,14,32]: given the incompleteness of available knowledge, we may need to use preferences and optimization criteria to select the best candidate solutions according to both qualitative and quantitative measures. Declarativity also represents a desirable feature because it allows the automatic encoding of medical knowledge, thus making the system easily extensible and medically sound.

Before we describe our reasoning tasks in detail, we want to recall some basic ASP definitions. In ASP a given problem is represented by a logic program whose results are given in terms of *answer sets*.

A logic program P is a finite set of rules r_i of the form

$$r_i : c \leftarrow a_1, \ldots, a_m, \text{not } b_{m+1}, \ldots, \text{not } b_n \ . \quad (1)$$

where each a_i, b_j, c is a *literal*, that is, an *atom* $p \in \mathcal{P}$ or its negation $\neg p$ for $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant n$, *not* is a logical connective called *negation as failure* and $n \geqslant m \geqslant 0$.

For a rule r_i as in (1), let $head(r_i) = \{c_1, \ldots, c_l\}$ be the *head* of r_i and

$$body(r_i) = \{a_1, \ldots, a_m, not\ b_1, \ldots, not\ b_n\}$$

be the *body* of r; and let $body^+(r_i) = \{a_1, \ldots, a_m\}$ and $body^-(r_i) = \{b_1, \ldots, b_n\}$.

Rules r_i with $head(r_i) = \emptyset$ are called *integrity constraints*, while if $body(r_i) = \emptyset$, we refer to r_i as *a fact*.

An interpretation I is represented by the set of atoms that are true in I. A *model* of a program P is an interpretation in which all rules of P are true according to the standard definition of truth in propositional logic. Apart from letting ',' stand for conjunction, this implies treating rules and default negation by not as implications and classical negation, respectively. Note that the (empty) head of an integrity constraint is false w.r.t. every interpretation, while the empty body is true w.r.t. every interpretation. Answer sets of P are particular models of P satisfying an additional stability criterion. Roughly, a set X of atoms is an answer set, if for every rule r_i of the program, $c \in X$ whenever $a_1, \ldots, a_m$ belong to X and no $b_{m+1}, \ldots, b_n$ belongs to X.

Formally, an *answer set* X of a program P is a minimal (in the sense of set-inclusion) model of

$$\{head(r_i) \leftarrow body^+(r_i) \mid r_i \in P, body^-(r_i) \cap X = \emptyset\}.$$

Although answer sets are usually defined on ground (i.e., variable-free) programs, the rich modeling language of ASP allows for non-ground problem encodings, where rules with variables (upper case names) are taken as a shorthand for the sets of all their ground instantiations. Grounders, such as *gringo* [2] and *lparse* [3], are capable of combining a problem encoding and a problem instance (typically a set of ground facts) into an equivalent ground program, which can then be then processed by one of the available ASP solvers. In our implementation of SINDI, answer set programs are grounded using Gringo [9] and interpreted using the Clasp [8] solver.

For extending the expressive power of logic programs, we take advantage of three kind of rules [32].

The first type is the *cardinality rule*, which is of the form:

$$r_i : h \leftarrow l\ \{a_1, \ldots, a_m\}\ u \tag{2}$$

where $C = l\ \{a_1, \ldots, a_m\}\ u$ is a *cardinality constraint* and l and u are two integers giving a *lower* and *upper* bound, respectively, on the number of satisfied literals within the constraint.

For a cardinality constraint C as in the body of (2), we let $lit(C)$ denote its set of literals $\{a_1, \ldots, a_m\}$ and let $lb(C) = l$ and $ub(C) = u$. C is *satisfied* by a set of literals S, if

$$lb(C) \leqslant |lit(C) \cap S| \leqslant ub(C)\ .$$

Whenever bound l or u is missing, it is taken to be 0 or $|lit(C)|$, respectively.

Rule 2 is interpreted as follows: if at least l and at most u literals in the set $\{a_1, \ldots, a_m\}$ are satisfied by a stable model S, than atom h should be in S.

The second type of rule is the *choice rule*, which is of the form

$$r_i : \{h1, \ldots, h_k\} \leftarrow body(r_i). \tag{3}$$

which implements a nondeterministic choice over the atoms in $\{h1, \ldots, h_k\}$ whenever $body(r_i)$ is true in the stable model. That is, the rule generates solutions including any number of atoms among $\{h1, \ldots, h_k\}$.

The third type of rule is the *weight rule*, which is of the form

If we associate a real-valued coefficient, i.e., a weight to each atom in the cardinality constraint, we obtain a generalization of rule 2, called *weight rule* of the form

$$r_i : h \leftarrow l\ \{a_1 = w_{a_1}, \ldots, a_m = w_{a_m}\}\ u \tag{4}$$

where l and u are again two integers giving a *lower* and *upper* bound, respectively, on the sum of the weights of satisfied literals within the constraint.

The idea is that a stable model satisfies the constraint if the sum of the weights of the literals satisfied by the model is between l and u. This means that the head of rule 4 is included in a stable model S if $l \leqslant \Sigma_{a_i \in S}\ w_{a_i} \leqslant u$. Whenever bound l or u is missing, it is taken to be $-\infty$ or $+\infty$, respectively. Answer sets of programs with cardinality and weight constraints are defined as in [23].

Another interesting feature of the ASP formalism is the possibility of expressing both quantitative and qualitative preferences. In the first case, we can use weight rules while for qualitative preferences (used in selecting the preferred action to be performed as a result of the application of a feedback policy) we intro-

[2] *A user's guide to gringo, clasp, clingo, and iclingo.* http://potassco. sourceforge.net

[3] Lparse 1.0. http://www.tcs.hut.fi/Software/smodels/lparse.ps.gz

duce Ordered Disjunction in the head of rules. We recall the basic definitions of Logic Program with Ordered Disjunction (LPOD), as given in [4] and [5].

Given an alphabet $\mathcal{P}$ of propositional symbols, an LPOD-program is a finite set of LPOD-rules of the form

$$c_1 \times \cdots \times c_l \leftarrow a_1, \ldots, a_m, not\ b_1, \ldots, not\ b_n. \quad (5)$$

where each a_i, b_j, c_k is a *literal*, that is, an *atom* $p \in \mathcal{P}$ or its negation $\neg p$ for $0 \leqslant i \leqslant m$, $0 \leqslant j \leqslant n$, and $0 \leqslant k \leqslant l$.

The "non-standard" part of such a rule is the *ordered disjunction* $c_1 \times \cdots \times c_l$ constituting its head. Given that the body literals are satisfied, its intuitive reading is:

- if possible c_1, but if c_1 is impossible, then c_2,
- $\ldots$,
- if all of $c_1, \ldots, c_{l-1}$ are impossible, then c_l.

Each c_k stands for a *choice* as a result of the application of rule (5). Note that the "$\times$" connective is allowed to appear in the head of rules only; it is used to define a preference relation that allows to *select* some of the answer sets of a program by using ranking of literals in the head of the rules, on the basis of a given strategy. The semantics of an LPOD program is then given in terms of a preference criterion over answer sets.

Answer sets of LPOD programs are formally defined as in [30].

4. Intelligent monitoring and assessment of well-being

We refer to an *intelligent monitoring system* as a monitoring system that is able to (i) reason about gathered data providing a context-aware interpretation of their meaning and (ii) support understanding and decision.

In the SINDI system, results of context-aware interpretation of gathered data are used to predict and explain possible evolutions of the person's health state in terms of functional disabilities, dependency in performing daily activities and risk assessment, as well as to identify correct interaction patterns [2,18]. In this section we want to focus on how the system reasons about incomplete and potentially inconsistent sensor data to contextualize them and use them in supporting intelligent monitoring.

4.1. Data collection and aggregation

First of all, reasoning about gathered data is used to understand what the person is doing in terms of movements, and to localize the person. Some data aggregation (fusion) is already performed at the feature extraction level with statistical algorithms, e.g. particle filters. Data can still be imprecise, even after this aggregation process. The expressive power of ASP is used at this stage to disambiguate unclear situations (e.g. where the person is) using commonsense, defaults, nondeterministic choices and preferences to select the best candidates in the space of the solutions.

The continuous measurements provided by sensors are stored in the database of SINDI with the results of reasoning. In the actual implementation, data are aggregated and extracted every hour to be processed by the logic inference engine. Discrete time is in seconds and the new data are extracted when values change beyond a given threshold. As an example, the light is aggregated as the average light in a room with a value $V \in \{0..500\}$, mapped in the following thresholds V_t according to the value of V:

$$\begin{array}{lll} V_t = dark & if\ V & \in \ \{0..125\} \\ V_t = shadow & if\ V & \in \ \{126..250\} \\ V_t = clear & if\ V & \in \ \{251..375\} \\ V_t = bright & if\ V & \in \ \{376..500\} \end{array}$$

A logic fact of the form

$$attribute\_env(ambientLight, Room, V_t, T)$$

is generated whenever the new light value is mapped to a different threshold. We are currently working on a wrapper that will make it possible to use the Clasp solver as a permanently running API that we can feed with aggregated sensor data as soon as they are available. This will make the reasoning process faster.

The high level representation of the collected data is automatically mapped into logic predicates resulting in a set of facts which is combined with the ASP logic program. This high level representation complies with the context model described in Section 2.

4.2. Context interpretation to support health assessment

As mentioned in the previous sections, commonsense reasoning is used in SINDI to supplement the description of the context when there is missing information. As an example, inertia law on the relation $in()$ can be used to detect where objects are at a given time, as follows:

```
out(Obj,Room,T) :- in(Obj,R1,T), R1!=R.
in(Obj,Room,T)  :- in(Obj,R,T-1),
                   not out(Obj,Room,T).
```

Interaction with multiple objects[4] can be consistently interpreted by generating all possible alternatives: the interaction with two distant objects at the same time puts the person in different locations, unless one of them is supported by additional evidence; a consistency constraint can be used to specify that options should be treated separately (one location for each time step), as follows:

```
n_attribute_obj(Name,Obj,V,T) :-
            attribute_obj(Name,Obj,V1,T),
            V1!=V, range(Name,V).

attribute_obj(Name,Obj,V,T)   :-
            attribute_obj(Name,Obj,V,T-1),
            not n_attribute_obj(Name,Obj,V,T),
            attribute_def(Name,Obj), time(T),
            range(Name,V).

% possible locations
change_val(Obj,T) :-
            attribute_obj(Name,Obj,V1,T-1),
            attribute_obj(Name,Obj,V,T), V!=V1.

poss_in(R,A,T)    :- change_val(Obj,T),
                     in(Obj,R,A,T).

% generate possibilities
{person_in(R,A,T) : poss_in(R,A,T)}.

% constraint on locations
:- person_in(R,A,T), person_in(R1,A1,T), A!=A1.
```

Besides localization, context interpretation is also a crucial phase in understanding basic behaviours that may be important for health assessment, such as night activity. Specific sequences of changes in the attributes of objects can help characterizing what is happening. As an example, a change in bed's load sensors followed by lights off in the bedroom may indicate that the person went to sleep, while a change in bed's and carpet's load sensors followed by switching the lights on in a room that is not the bedroom, indicates that the person got out of bed, as follows:

```
in_bed(T)  :-
      person_in(bedroom,bed,T),
      attribute_env(luce,bedroom,bedroom,dark,T),
      attribute_obj(state,imud,off,T),
      filteredLoad(bed,V,T), V>0.
out_bed(T) :-
      person_in(bedroom,bed,T1),
      filteredLoad(carpet,V,T), V>0, T1<T.

localized(T) :- presence(R,A,T).
awake(T)     :- not in_bed(T), localized(T).
awake(T)     :- out_bed(T).
bad_sleep(T) :-
       attribute_obj(loadVolatility,bed,N,T),
       in_bed(T), N!=stable.

sleep_interrupt(T) :- in_bed(T1), awake(T), T1<T.
back_to_bed(T)     :- sleep_interrupt(T0), T0<T1,
                      awake(T1), in_bed(T), T1<T.

nightstart(T) :- in_bed(T).
nightend(T)   :- sleep_interrupt(T), T<T1
                 not back_to_bed(T1), time(T).
```

In this paper, we do not focus on the way SINDI deals with emergencies, but we want to mention that they are introduced in our reasoning model in a straightforward way: the condition indicating an emergency (such as the temperature going beyond secure threshold or a fall detected according to the inertial sensors) can be specified as a logic rule, and the predicate representing the emergency can be added as an exception in the body of rules illustrated above, using *not*[5]. In general, results of the context interpretation process are used for the (absolute and differential) evaluation of (i) significant aspects of the patient's quality of life (referred to as *indicators*) in terms of clinical profile, quality of the environment, quality of movements, quality of sleep and other daily behavior and (ii) a well-defined set of health-related factors (referred to as *items*) according to influences among them and values of related indicators.

The relation between items and indicators is the following: each indicator can contribute to the evaluation of one or more items when no direct evaluation of the item itself is available; one indicator may contribute to the evaluation of several items and one item can be evaluated through changes of several indicators; while indicators do not have any mutual dependency or correlation among them, items are correlated by dependency relations indicating how a change in the value of an item may impact values of other items (see Section 5 for details on how such dependencies are used for prevention of falls).

[4] In the simplest scenario, a person is supposed to interact with an object when the value of one attribute associated to the object changes. These changes are also propagated using the inertia law.

[5] Note that *not* in ASP is intended as *negation as failure*.

Table 5

Indicators of well-being evaluated through context-interpretation

Indicator Name	Description
dayActivity	Level of activity vs. inactivity period during the day
earlyNight	Quality of sleep in the early night hours (11 p.m. to 1 a.m.)
middleNight	Quality of sleep in the middle night hours (1 to 4 a.m.)
lateNight	Quality of sleep in the early morning hours (4 to 6 a.m.)
tempQual / *humQual* / *brightQual*	Quality of the environment w.r.t. temperature / humidity / light
...	...

Indicators used to assess the quality of life of the elderly have been identified with the help of domain experts and they can be evaluated through results of questionnaires/clinical tests or continuous monitoring supported by (i) data aggregation (e.g. quality of movement), (ii) specific inference rules (e.g. quality of sleep) and (iii) commonsense reasoning (e.g. quality of the environment). A reduced list of indicators evaluated by the system and encoded in form of logic facts is provided in Table 5.

At each reasoning cycle, indicators are evaluated as a result of the interpretation of context data, and their value is compared with the results of previous inferences, thus obtaining differential values. Admissible values for each indicator are part of the medical knowledge and are encoded in the knowledge base of the system; their differential evaluation has four possible outcomes: worsening, improvement, no substantial change, undefined.

Results of the absolute evaluation are defined over specific ranges according to the medical practice, which also guides the conditions used in logic rules for evaluation, and they are expressed as logic predicates of the form $obsInd(I, V, H)$, where H is a time stamp identifying the hour associated to the inference cycle the evaluation refers to[6].

If we consider again the night activity, context interpretation makes it possible to evaluate indicators *earlyNight*, *middleNight* and *lateNight* associated to the quality of sleep. The following encoding is related to *lateNight* according to the Hamiltonian Scale[7] for the score of *Insomnia Late*:

0 = No difficulty in sleeping

1 = Walking in early hours of the morning but goes back to sleep

2 = Unable to sleep again if he gets out of bed

The correspondent encoding in ASP is as follows:

```
sleep_break      :- sleep_interrupt(T).
back_sleep(T,T1) :- sleep_interrupt(T),
                    back_to_bed(T1), T<T1.

obsInd(lateNight,ok,0) :- not sleep_break,
                          not exception.
obsInd(lateNight,moderate,0)    :-
    sleep_interrupt(T), presence(R,A,T1),
    R!=bedroom, back_sleep(T,T2), T<T1, T1<T2,
    not obsInd(lateNight,consistent,0),
    not exception.
obsInd(lateNight,consistent,0) :-
    sleep_interrupt(T), not back_sleep(T),
    not exception.
exception :- fall_detected.
```

Given the difficulty of the evaluation when there is no human observer involved, what we obtain from the reasoning process described so far is a *partial* evaluation. Exceptions specific to the patient can be introduced in the encoding (such as the triggering of the evaluation rules only when there are no ongoing treatments that may impact the evaluation). The following step is that of using static dependencies and influence relations between evaluated items in order to predict the risk of worsening for items which have not been evaluated. This process which we call *prediction* is an off-line reasoning mechanism that suggests the caretaker which are the risky items one has to pay attention to, given a partial evaluation. At the same time, results of the partial evaluation are used by the system to determine how to react to certain conditions (*prevention*). In the next section we illustrate SINDI's reasoning tasks aimed at prediction and prevention in home healthcare.

5. Context-aware prediction and prevention

A careful analysis of health care in home settings suggests that health-related aspects that are crucial for elderly care (referred to as *items*) can be classified into three classes [17]:

[6] Time stamp $H = 0$ is associated to the hour being evaluated, while timestamps $H > 0$ refers to previous inference steps: the highest H, the oldest the hour.

[7] http://healthnet.umassmed.edu/mhealth/HAMD.pdf

- *functionalities*, evaluated in terms of functional disabilities including physical and cognitive functional aspects of elderly behavior;
- *daily activities*, evaluated in terms of level of dependence in performing daily activities;
- *risks*, representing complex aspects usually involving functional disabilities, level of dependence in daily activities and the clinical setting.

Items have been identified according to the medical practice in health assessment of the elderly [7] and encoded in our declarative framework. An additional class of items represents the *state* of the person and the *state* of the environment in our context-aware model of health. This classification is important for clinicians because it makes it possible to keep the structure of knowledge they are used to and thus it makes it easier to investigate inference results.

Starting from these considerations, we characterize SINDI's reasoning tasks aimed at prediction and prevention in our context-dependent scenario. We refer to *prediction* as one or more (consistent) guesses that a healthy condition might change into a risky condition. Prevention is referred to as all those interventions that may keep health changes within safe boundaries. Prediction and prevention are strictly related: when risky conditions are identified by the system as a result of prediction, appropriate feedback actions are selected and performed for prevention. We focus on the combination of pervasive monitoring and efficient reasoning to support *prediction* and *prevention* in the context of a logic-based approach that tackles both physical and cognitive impairments.

In both prediction and prevention, context-awareness is a desirable property since reasoning support should consider not only static clinical knowledge, but also user-specific needs as well as the evolving state of the patient and of the environment. We use non-monotonic logical reasoning to summarize and correlate sensor data in a consistent interpretation of the situation, taking into account knowledge about elderly care, state of the environment, state of the person monitored (localization, movement), clinical profile information, and results of previous inferences.

The evaluation process results in incomplete information about items' values, represented by a partially labeled graph where nodes are items and oriented arcs are influences of value changes of an item on the evaluation of other items. The following reasoning step is in charge of predicting possible health changes for unlabeled items; intuitively this is done by computing all possible consistent labellings of the graph for the missing information according to SINDI's logic model of dependencies between items.

Arcs representing *dependencies* connecting items are similar to arcs going from indicators to items. An item may influence/be influenced by several items.

SINDI's knowledge model allows different kinds of dependencies among a source (item I_i or indicator Ind_i) and a target (item I):

1. strictly negative/positive influence of item I (resp. of indicators Ind_i) on item I_1;
2. strictly negative/positive inverse influence of item I (resp. of indicators Ind_i) on item I_1;
3. directly proportional influence of item I (resp. of indicators Ind_i) on item I_1;
4. inversely proportional influence of item I (resp. of indicators Ind_i) on item I_1.

As a result of prevention, the logic program may yield different solutions according to which of the dependency relations prevail over the others. From these results we extract labellings that are coherent in all solutions or the ones that maximize the sum of weights of the dependency arcs that determine the labellings [21].

The reasoning task also provides explanation (in terms of alternative paths in the dependency graph) for the inferred guesses. As a result, caregivers are supported in understanding how and why the health state of a person could evolve in a given direction. We also deployed a graphical interface to let caregivers access data intuitively, but we still do not have sufficient test data to give a fair evaluation of the caregiver interface.

The logical framework of Answer Set Programming (ASP) [10] is well suited to deal with such a complex knowledge representation and reasoning task, in that it overcomes most of the limitations of previous logic programming systems. Compared to pure statistical approaches, logic inference based on ASP is highly expressive and computationally more efficient because it can deal with first-order representations, which are much richer than the propositional ones characterizing probabilistic inference. Furthermore, ASP can deal with incomplete information and commonsense reasoning using defaults. Cardinality and weight constraints together with program optimization techniques can also be used to model different degrees of uncertainty [9,14,32]. We use Clingo as the ASP reasoning engine[8].

[8] http://potassco.sourceforge.net/

In the remainder of this section we focus on how we represent appropriate knowledge used by the system to react to specific situations and support prevention.

Since we believe feedback represents the key for effective preventive interventions, an additional part of the domain knowledge of SINDI is related to the representation of admissible feedback from the system to the person monitored. Furthermore, given that each person has a different clinical history of cognitive decline and reacts in different ways to external stimuli, it is extremely important to select the most appropriate feedback according to the context.

The system can provide feedback in five different ways:

- *suggestions* according to the medical practice and the results of the prediction reasoning task;
- *alerts* when the system identifies behaviors or situations that are potentially dangerous according to the results of the prediction reasoning task;
- *alarms* when specific environmental or clinical conditions are detected;
- *notifications* when the system receives new input or terminates the inference process;
- *reminders* according to an agenda.

The main difference between a suggestion and an alert is that the second is triggered by the identification of a specific behavior and may generate an immediate action as output (e.g. a blinking light to indicate that there is a call), while the first is purely based on the medical knowledge encoded in the system and gives a report as output. Alarms also generate an action but usually need an immediate response (e.g. a call to the caregiver when a fall is detected).

In our first specification reminders do not include support on how to perform complex activities as in [3, 25]. The system deals only with simple reminders according to an agenda.

Independently of how it is delivered, a feedback action can be related to:

- the environment: making the environment safer and of better quality, improving interaction with the environment, e.g. a phone call that is not acknowledged by the patient can trigger actions like reducing the volume of the TV or blinking a light;
- the user's behavior: suggesting how to modify habits when the health assessment indicates risky conditions or providing reminders according to an agenda;
- the clinical setting: consulting a doctor, suggesting a more accurate test, reviewing a therapy, reminding medical appointments, and so on.

The combination of inference results (prediction) and context-related knowledge about the person and the environment is used to determine (i) *what* should be provided as feedback, (ii) in *which form* and (iii) *when*.

The content of the feedback is determined according to the medical literature (evidence-based studies) and encoded in the system (see Section 6 for an example of fall prevention).

The most appropriate form of feedback is inferred by the system on the basis of the results of prediction and context-related information about the patient and the environment: the same feedback can be provided in different forms (and at different times).

A feedback can be provided as soon as it is inferred or at a later time. Alarms are usually immediate, while other forms of feedback can be performed immediately or at a later time according to:

- triggers: pushing a button at a specific time or when particular conditions hold;
- user/caregiver preferences: qualitative ordering to identify more urgent/important suggestions according to the environmental/personal context setting and the form of feedback;
- static ordering: certain forms of feedback may have higher priority than others, simply because of their nature; similarly, some communication patterns can be preferred to others on the basis of clinical settings.

When the system determines a set of feedback actions that should be performed at a given time, they are qualitatively analyzed in order to infer which action is more urgent. Each feedback action can be associated to a list of possible reactions of the patient. Simple reactions can be monitored through sensors, while more complex reactions can be inferred combining sensors and reasoning.

A reaction to a feedback, when detected, is logged to be used at a later time. Exploring this history, caregivers can improve the way feedback actions are performed and identify the most effective communication patterns.

Learning interaction patterns is an interesting issue [28], but we do not tackle it in this paper. Given that we consider different forms of feedback, an appropriate reward function should be identified in order to take into account how the combination of different feedback actions and communication patterns impact the quality of life of the person monitored.

6. A case study: Prediction and prevention of falls

Falls are the top cause for loss of independence which then leads to a lower quality of life. It has been shown that hip fractures continue to generate significant costs throughout the first-year period after discharge. Costs associated with the treatment of hip-fracture patients are about three times greater than those resulting from the treatment of age and residence-matched controls without a fracture [11]. As a matter of fact, causes of falls in elderly adults are quite diverse, the largest culprits being: environment-related (31%), gait/balance disorders (17%) and dizziness/vertigo (13%) [27]. Identifying risk factors and initiating preventive interventions can substantially lower the probability of a fall. It is therefore very advantageous both for the patient's quality of life and for the minimization of the costs to devise systems that are able to warn caregivers and patients that action has to be taken well before a fall happens.

It is hard to identify a single specific cause for falling, as falls usually depend on multiple interrelated factors [27]. Some of them are mainly related to clinical aspects (metabolism, nervous systems, anaemia, hypothyroidism, osteoporosis and so on), while others are more concerned with causal relations among aspects (quality of the environment and health-related items) whose worsening may predispose to risk of falls.

We are interested in the latter set of causes, and want to illustrate how the SINDI system reasons about dependencies to support falls prediction and prevention. Several assessment tools have been proposed and used to evaluate the risk of falls, most of them impossible to apply without the help of a skilled human observer [24,35,40]. The reasoning component of our system partially assesses the risk of falls according to the Tinetti Performance-Oriented Mobility Assessment [35] scale, as it is one of the standard tools in our country for assessing mobility dysfunctions in the elderly. Since balance and gait assessment using sensor data analysis are sometimes non-definitive, they are combined with other information about the environment, the clinical profile and evidence-based medical knowledge derived from experimental trials [6,22,27,34].

6.1. Knowledge-based model and reasoning tasks

In the remaining part of this section we describe the reasoning mechanism of SINDI applied to prediction and prevention of falls. The Knowledge-based model is as described in Section 2 and the reasoning tasks are (i) prediction intended as the identification of possible health evolutions, and (ii) prevention intended as the action of providing appropriate feedback as a result of prediction, and observing reactions to that feedback.

6.1.1. Fall prediction

In the SINDI system, health-related items that influence the risk of falls are illustrated in Table 6. We recall that indicators associated to items are periodically evaluated and labelled as worsening, improving, being stable or undefined, according to data gathered by the sensors, test results and logic-based context interpretation and evaluation.

In order to identify a problematic item, the reasoning process takes into account the evaluation of all indicators associated with that item and some additional knowledge about specific pathologies and drugs that can influence the evaluation.

Once problematic items have been identified, the reasoning system analyses item-related dependencies to infer possible evolutions of the general health state and identifies items that are not yet problematic, but are potentially at risk. A subset of dependencies among items that are relevant to the risk of falls can be seen in Fig. 2. In this way, the results of reasoning are not purely clinical, but they are related to the context in

Table 6

Items and indicators used for prediction

Item	Indicators used for evaluation
Mobility	# steps, walking time
Balance	sitting down, standing up, equilibrium when sitting/standing, test results
Gait	walking/turning speed, step length, test results
Nutrition	weight loss, diet, test results
BMI	weight
Vision	optometric tests, lights usage
Hearing	audiometric tests
Sleep	night activity, hours of sleep
Environment	lightening, humidity, temperature, barriers
Clinical profile	# diseases, # drugs, age, gender

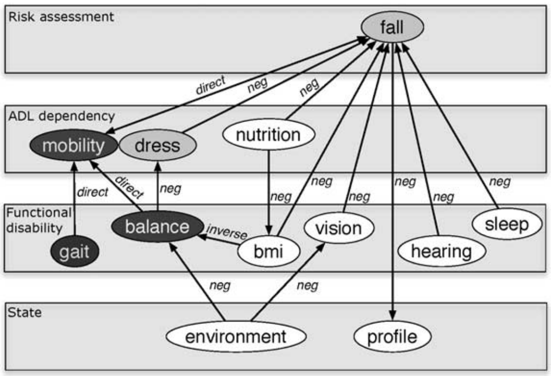

Fig. 2. Results of prediction for the example in Section 6.2.

which they have been observed and to the evolution of the person's health state.

To sum it up, fall prediction is achieved through the following steps:

- sensor data are aggregated and interpreted in order to evaluate relevant indicators;
- values of indicators are used to identify problematic items;
- dependency relations among items are used to identify items at risk;
- following causal dependencies backwards, the system is able to explain why an item is tagged as being at risk.

6.1.2. *Fall prevention*

Results of prediction are used by SINDI to provide feedback aimed at prevention. Research on the impact of feedback in reducing the risk of falls has shown that people informed that they are at risk of falls are more likely to make changes in their habits and in the environment they live in [22]. We identified a well known set of aspects that can significantly contribute to fall prevention among persons that are at risk. These aspects are summarized in Table 7.

Possible forms of feedback are indicated as S (suggestion), AA (alarm), A (alert), N (notification), R (reminder) and they can be related to:

- the environment: how to make the environment safer (e.g. suggestions about lighting, detection of events that the patient has not noticed, and so on);
- correct behavior: which healthy habits should be practiced and which potentially risky behaviors are to be avoided;
- clinical actions: consult a doctor, suggest a more accurate test, review medication, provide reminders according to an agenda, and so on.

It is interesting to notice that, according to [34,36], falls prevention strategies targeting several risk factors concurrently are more likely to have a greater impact in reducing falls than strategies that target only one risk factor.

Knowledge contained in Table 7 is represented as logic predicates of the form $possible\_form(Output, Form)$ associating the content of a feedback to its possible forms. In order to identify which feedback to provide, in which form and when, the reasoning component applies a set of rules we refer to as a *feedback policy*. Rules of a feedback policy can be of six different types:

Event-triggering rules trigger a specific form F as a candidate for a feedback output X when events

Table 7

Classification of feedback outputs for fall prevention

Class	Output	Form	System Action
Environment	do not walk in the dark	S, A, N	turn lights on
	check temperature	S, A, N	vocal warning
	incoming call	S, A, AA, N	blinking lights
	remove clutter	S, A, N	vocal warning
	emergency	AA	call 911
Behaviour	keep active	S, A, N, R	propose exercises
	stand up slowly	S, A, N	vocal warning
	fix carpets	S	vocal warning
	use chair to get dressed	S	vocal warning
Clinical actions	review drugs	S	on screen
	see a specialist	S, R	on screen

(S) suggestion (AA) alarm (A) alert (N) notification (R) reminder

Table 8

Mapping policy rules into ASP

Rule Description	*ASP Encoding*
event-triggering rule	$r_{k1} : feedback\_form(X, F) \leftarrow obs(e_1), \dots, obs(e_n), poss\_form(X, F),\ [C].$
feedback selection	$r_{k2} :\ min\{sel\_form(F_1), \dots, sel\_form(F_k)\}max\ \leftarrow$ $feedback\_form(X, F),\ [C].$
default preference rule	$r_{k3} :\ sel\_form(F_i) \times sel\_form(F_j)\ \leftarrow\ not\ exception,\ [C].$
exceptions to default	$exception \leftarrow [C].$

$E_1, \dots, E_n$ are detected and a boolean condition C holds:

$$E_1, \dots, E_n\ \textbf{triggers}\ feedback\_form(X, F)\ \textbf{if}\ C \quad (6)$$

Multiple choice rules express the fact that one of all possible forms F_i for a given feedback output X can be selected at each inference cycle:

$$1\{select\_form(X, F_i) : feedback\_form(X, F_i)\}1 \quad (7)$$

Default ordering rules are binary relations expressing the fact that, in general, a form F_1 for a given feedback output X is preferred to the form F_2 for the same feedback output, where $>$ is a transitive ordering relation:

$$select\_form(X, F_1) > select\_form(X, F_2) \quad (8)$$

Exceptions to default ordering are used to make a default ordering rule not applicable when a boolean condition C holds:

$$exception(X)\ \textbf{if}\ C \quad (9)$$

Event-Condition-Action (ECA) rules express the fact that an action A should be performed by the system when a given feedback output X of the form F is the chosen candidate and a condition C holds:

$$select\_form(X, F)\ \textbf{causes}\ A\ \textbf{if}\ C \quad (10)$$

Consistency rules express the fact that some actions $A_1, \dots, A_n$ cannot be executed together when boolean condition C holds, and there is a preferential relation on which action should be dropped:

$$never\ A_1 \times \dots \times A_n\ \textbf{if}\ C \quad (11)$$

All these high level rules are automatically mapped into ASP as illustrated in Table 8, while mapping for rules of the form in Eqs (10) and (11) is done in a similar way as in [19,20].

The rule-based specification illustrated above is a compact way to express:

- which feedback outputs are triggered by a given set of events;
- non-deterministic choice among alternative forms for a feedback output;
- a static ordering on possible forms for the same feedback output and exceptions to this ordering;
- actions associated to a feedback output of a specific form and a preference relation over conflicting actions.

The result of the enforcement of a feedback policy is a consistent set of actions to be performed by the system.

The application of a feedback policy may yield several solutions representing acceptable communication patterns. In the current specification, one of the solutions that minimizes the number of interruptions is selected. It is interesting to study methods that enable the system to learn the most effective communication patterns, but the complexity of this domain makes it harder than in contexts where only user preferences matter [38].

6.2. Example

In the following example we want to capture one possible concrete scenario to illustrate how the reasoning component works in prediction and prevention tasks.

In this scenario, we track a hypothetical patient monitored by SINDI, called Eve. When Eve wakes up, the data aggregation module and the reasoning component of SINDI track her getting out of bed, movements and location around the house. After a long period sitting on the sofa and watching TV in the morning, Eve walks to the kitchen to prepare some food. The evaluation of her gait indicates that something has changed, since walking speed is reduced due to the long inactivity period. Trying to get things out of a cupboard, Eve slightly injures her back. This is noticed by the system because her subsequent sitting-down movements are performed with more difficulty.

Functional dependencies are evaluated every hour, thus SINDI identifies gait, balance and mobility as problematic after analyzing indicators like walking speed and quality of sitting action in the time interval under investigation. Given that gait and balance problems directly affect mobility and that a reduced mobility may have a negative impact on the risk of falls, SINDI predicts a possible worsening of the risk of falls. Balance also influences dependency in getting dressed; though we have no indicators-based evidence of it, the prediction task marks dependency in this activity as being potentially at risk. Results of this step of the inference process are illustrated in Fig. 2.

A set of event-triggering rules (6) identifies *potential* feedback in form of suggestions: stand up slowly, use a stable chair to get dressed, don't stay inactive for too long during the day, keep stairs and walk areas clear of clutter. An ECA rule (10) indicates that feedback outputs in form of suggestions and notifications should be provided as a report at the end of the day. This holds unless a form of higher priority is triggered for the same feedback.

Later, the quality of the environment with respect to light is marked as decreased. According to the prediction task, a wrong use of lights may indicate visual problems, thus a possible disability in vision is inferred and added among possible causes for the prediction of an increase of the risk of falls. Through the specification of an appropriate event-triggering rule (6), another suggestion is added to the list: do not walk across a dark area.

In the following inference cycle, context interpretation reveals that Eve walked through areas that were not properly lighted. At this point, an event triggering rule indicates that the same feedback output should now be treated as an alert. Given that there is a default ordering rule giving alerts higher priority than suggestions, alert becomes the preferred form for the feedback "do not walk across a dark area".

While suggestions are usually included in a report, alerts are usually associated with a specific action to be performed. If there are no other conflicting actions inferred by the system, the feedback is provided as indicated by the corresponding ECA rule, in a way that minimizes interruptions.

This example shows how inference results predicting potential risks are used to identify the appropriate feedback for prevention, and how policy rules make this list dynamic according to how the prediction and the context evolve. In this way, SINDI can provide the most appropriate feedback at the right time. In fact, if the wrong use of lights was not detected, a feedback related to the use of lights would have been a suggestion in the final report, rather than an alert.

In the next day, the impact of feedback and the way Eve reacts to them are monitored: in case Eve reacts to one of the feedback outputs (e.g. she becomes more active), a policy rule of the form described in (6) can generate a Notification that is added to the daily report. Notifications are used to give evidence of (i) whether Eve has followed or not feedback outputs and (ii) how much does preventive intervention impact the risk of falls. This makes it possible to keep track of the whole cycle (feedback, reaction to feedback when available, impact on possible evolutions of the health state) and it can be a source of data for tests and trials to identify the correct intervention for a more general class of patients.

7. Preliminary evaluation

The main contribution of our solution is that data are interpreted by a component that is able to draw from a knowledge database and make complex inferences. Our approach to context-aware monitoring has the following desirable characteristics to support healthcare and well-being:

- declarative context representation and interpretation make it possible to cope with an assortment of patient conditions under different settings;
- prediction of situations that can cause drastic changes for the worse of the quality of living make it possible to identify preventive strategies to maintain autonomy and independence for a longer time;

– continuous monitoring makes it possible to automatically collect a massive amount of data that can be used to identify interesting case studies.

It is very hard to evaluate the performance of SINDI in its entirety without extensive field deployment and analysis. We are working towards this goal but do not yet have hard results from field experiments that are long enough and diverse enough to yield scientific significance.

However, we tested the correctness of inference results concerning context interpretation based on plausible sensor data. To do that, we generated large data sets simulating several years of the household activities of an agent-modeled person using *Repast Simphony (Repast S)*, a free and open source agent-based modeling toolkit. The simulation scenario is specified in an XML file describing rooms, areas, properties of cells, position of sensors, light and heat sources, status of windows and light switches.

The data-generation is driven by two different event schedulers that are used to: (i) simulate the behaviour and the habits of the person, (ii) simulate light, temperature and humidity evolution during the day.

In order to simulate the behaviour and the habits of a person living and moving in a realistic home environment, we defined an XML-based file that contains a detailed list of high-level *Activities* that the *Person* agent is required to execute during the data generation; activities are composed of one or more *ActionSets*, which are composed of several low-level *Actions* that indeed specify what the Person agent should do.

The second event scheduler used by the data generation tool is the one controlling the evolution of the natural light during the simulated day and the propagation of temperature, humidity and light variations in the simulation environment.

Using such data generated in a complete simulation scenario, we did several tests on the WSN and on the inference engine[9], mainly concerning the localization algorithm: the correct location in the grid is identified 70% of the time without further filtering techniques, and the average distance of the solution from the real position is between 25 and 50 centimeters.

The reasoning processes for context interpretation and evaluation takes advantage of the non-monotonic nature of ASP. If we consider localization, results are obtained quite fast on a grid of 50×50 cells with 500 time steps. However, when the size of the ground instance becomes bigger (e.g. for planning and scheduling), we believe that the integration of ASP reasoning with constraint solving techniques [16] could make context interpretation from sensor data more efficient and we plan to investigate it. This of course does not include situations in which emergencies arise, since they are detected almost immediately.

Reasoning tasks for prediction rely on the computational power of ASP: a complex reasoning process takes into account the whole of dependencies to find maximal consistent labellings; local explanation can consider an arbitrary class of items to find subsets of dependency arcs that represent a minimal common explanation, obtained by using qualitative and quantitative optimization techniques. Tests on random instances[10] showed that when a solution exists, the prediction task produces an average of 60% consistent guesses for the unlabeled items even with less than 10% initial information.

Prevention is based on a compact representation of feedback policies that are automatically mapped into ASP and enforced by applying consistency mechanisms and adaptive optimization strategies taking into account contextual settings.

Given the high variability among trials and studies addressing prediction and prevention issues, it is still difficult to extract a coherent picture of what leads to disability and to develop coherent prevention strategies. In this respect, our system has the potential to automatically collect a massive amount of data in order to evaluate context-related prediction patterns and effective communication strategies for prevention.

There are several issues that have to be considered and which need further investigation. Among them, we mention the fact that extending the Knowledge Base of the system with new evidence-based medical knowledge in a modular and consistent way is a challenge, and appropriate interaction paradigms targeted to the elderly should be designed and developed.

References

[1] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, pages 263–277, 2007.

[9] We evaluated ASP programs by using *Gringo* [9] and *Clasp* [8], which supports constraints, choice rules and weight rules [23].

[10] Randomness is obtained by generating partial consistent colourings of the dependency graph.

[2] R. Bisiani, D. Merico, A. Mileo, and S. Pinardi. A logical approach to home healthcare with intelligent sensor-network support. *The Computer Journal*, page bxn074, 2009. doi: 10.1093/comjnl/bxn074.

[3] J. Boger, J. Hoey, P. Poupart, C. Boutilier, G. Fernie, and A. Mihailidis. A planning system based on markov decision processes to guide people with dementia through activities of daily living. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):323–333, 2006.

[4] G. Brewka. Logic programming with ordered disjunction. In *In Proc. of AAAI-02*, pages 100–105. AAAI Press, 2002.

[5] G. Brewka, I. Niemelä, and T. Syrjänen. Implementing ordered disjunction using answer set solvers for normal programs. In *Logics in Artificial Intelligence – Journées Européennes sur la Logique en Intelligence Artificielle*, pages 444–455, Cosenza, Italy, 2002. Springer.

[6] B.R. Connell and S.L. Wolf. Environmental and behavioral circumstances associated with falls at home among healthy elderly individuals. *Archives of Physical Medicine and Rehabilitation*, 78:179–186, 1997.

[7] K.C. Fleming, J.M. Evans, D.C. Weber, and D.S. Chutka. Practical functional assessment of elderly persons: A primary-care approach. *Mayo Clinic*, 70:890–910, 1995.

[8] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp: A conflict-driven answer set solver. In *Ninth International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 260–265. Springer-Verlag, 2007.

[9] M. Gebser, T. Schaub, and S. Thiele. Gringo : A new grounder for answer set programming. In *LPNMR*, pages 266–271, 2007.

[10] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP 88*, pages 1070–1080. MIT Press, Massachussets, USA, 15-18 August 1988.

[11] P. Haentjens, G. Lamraski, and S. Boonen. Costs and consequences of hip fracture occurrence in old age: An economic perspective. *Disability & Rehabilitation*, 27(18-19):1129–1141, 2005.

[12] K.Z. Haigh and H. Yanco. Automation as caregiver: A survey of issues and technologies. In *Proc. of the AAAI-02 Workshop "Automation as Caregiver"*, pages 39–53. AAAI, 2002.

[13] R. Kadouche, B. Abdulrazak, M. Mokhtari, S. Giroux, and H. Pigot. Semantic matching framework for handicap situation detection in smart environments. *Ambient Intelligence and Smart Environments*, (3):223–234, 2009.

[14] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7 (3):499–562, 2006.

[15] L. Liao, D. Patterson, D. Fox, and H. Kautz. Behavior recognition in assisted cognition. In *Proc. of the AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*, 2004.

[16] V.S. Mellarkod, M. Gelfond, and Y. Zhang. Integrating answer set programming and constraint logic programming. *Ann. Math. Artif. Intell.*, 53(1-4):251–287, 2008.

[17] D. Merico, A. Mileo, S. Pinardi, and R. Bisiani. A Logical Approach to Home Healthcare with Intelligent Sensor-Network Support. *The Computer Journal*, page bxn074, 2009.

[18] A. Mileo and R. Bisiani. Context-aware prediction and prevention to extend healthy life years:preventing falls. In *Proc. of the IJCAI Workshop on Intelligent Systems for Assisted Cognition, to appear*, 2009.

[19] A. Mileo and T. Schaub. Qualitative constraint enforcement in advanced policy specification. In *Proceedings of ECSQARU 2007*, pages 695–706. Springer-Verlag, 2007.

[20] A. Mileo, A. Provetti, and E. Bertino. PDL with preferences. In *Proceedings of POLICY 2005*, pages 213–222. IEEE, 2005.

[21] A. Mileo, D. Merico, and R. Bisiani. Reasoning support for risk prediction and prevention in assisted living. *Theory and Practice of Logic Programming*, 2009. Submitted.

[22] K.K. Ness, J.G. Gurney, and G.H. Ice. Screening, education, and associated behavioral responses to reduce risk for falls among poeple over age 65 years atending a communiti health fair. *Physical Therapy*, 83(7):631–637, 2003.

[23] I. Niemelä and P. Simons. Extending the smodels system with cardinality and weight constraints. In J. Minker, editor, *Logic-based artificial intelligence*, pages 491–521. Kluwer Academic Publishers, 2000.

[24] D. Podsiadlo and S. Richardson. The timed "up & go": A test of basic functional mobility for frail elderly person. *Journal of the American Geriatric Society*, 39(2):142–148, 1991.

[25] M.E. Pollack, L.E. Brown, D.Colbry, C.E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44 (3-4):273–282, 2003.

[26] A. Ranganathan and R.H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Comput.*, 7(6):353–364, 2003.

[27] L. Rubenstein. Falls in older people: epidemiology, risk factors and strategies for prevention. *Age Ageing*, (2):37–41, 2006.

[28] M. Rudary, S. Singh, and M.E. Pollack. Adaptive cognitive orthotics: combining reinforcement learning and constraint-based temporal reasoning. In *Proc. of ICML 2004*, pages 91–98. ACM, 2004.

[29] N.S. Ryan, J. Pascoe, and D.R. Morse. Enhanced reality fieldwork: the context-aware archaeological assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer Applications in Archaeology 1997*, British Archaeological Reports. Tempus Reparatum, 1998.

[30] C. Sakama and K. Inoue. An alternative approach to the semantics of disjunctive logic programs and deductive databases. *J. Autom. Reasoning*, 13(1):145–172, 1994.

[31] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proc. of Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, 1994.

[32] P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence Journal*, 138(1–2):181–234, 2002.

[33] T. Strang and C. Linnhoff-Popien. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004, Nottingham/England*, 2004.

[34] A.E. Stuck, J.M. Walthert, T. Nikolaus, christophe J. Bula, C. Hohmann, and J.C. Beck. Risk factors for functional status decline in community-living elderly people: a systematic literature review. *Social Science & Medicine*, 48(4):445–469, 1999.

[35] M.E. Tinetti. Performance-oriented assessment of mobility problems in elderly patients. *JAGS*, 34:119–126, 1993.

[36] T. Tse. The environment and falls prevention: Do environmental modifications make a difference? *Australian Occupational Therapy Journal*, 52:271–281, 2005.

[37] X.H. Wang, D.Q. Zhang, T.Gu, and H.K. Pung. Ontology based context modeling and reasoning using owl. pages 18–22, 2004.

[38] J.S. Weber and M.E. Pollack. Evaluating user preferences for adaptive reminding. In *Proc. of CHI 2008*, pages 2949–2954. ACM, 2008.

[39] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy. Distributed recognition of human actions using wearable motion sensor networks. *Ambient Intelligence and Smart Environments*, (2): 103–115, 2009.

[40] L. Yardley, N. Beyer, K. Hauer, G. Kempen, C. Piot-Ziegler, and C. Todd. Development and initial validation of the falls efficacy scale-international (fes-i). *Age and Ageing*, 34(6):614–619, 2005.