# A logical approach to home healthcare with intelligent sensor-network support



*Published version:*

DOI:10.1093/comjnl/bxn074



(Article begins on next page)

Advance Access publication on January 24, 2009
doi:10.1093/comjnl/bxn074

# A Logical Approach to Home Healthcare with Intelligent Sensor-Network Support

ALESSANDRA MILEO*, DAVIDE MERICO, STEFANO PINARDI AND ROBERTO BISIANI

*NOMADIS Research Lab, University of Milan-Bicocca, viale dell'Innovazione 10, I-20125 Milan, Italy*
**Corresponding author: alessandra.mileo@nomadis.unimib.it*

**This paper describes an intelligent home healthcare system characterized by a wireless sensor network (WSN) and a reasoning component. The aim of the system is to allow constant and unobtrusive monitoring of a patient in order to enhance autonomy and increase quality of life. Data collected by the sensor network are used to support a reasoning component, which is based on answer set programming (ASP), in performing three main reasoning tasks: (i) continuous contextualization of the physical, mental and social state of a patient, (ii) prediction of possibly risky situations and (iii) identification of plausible causes for the worsening of a patient's health. Starting from different data sources (sensor data, test results, inference results) the reasoning component applies expressive logic rules aimed at correct interpretation of incomplete or inconsistent contextual information, and evaluates correlation rules expressed by clinicians. The expressive power of ASP allows efficient enough reasoning to support prevention, while declarativity simplifies rule-specification and allows automatic encoding of knowledge. Preliminary evaluations show that the combination of an ASP-based reasoning component and a WSN is a good solution for creating a home-based healthcare system.**

*Keywords: knowledge representation; logic programming; wireless sensors networks; independent living*

*Received 29 February 2008; revised 17 December 2008*
Handling editors: Fariba Sadri and Kostas Stathis

## 1. INTRODUCTION

In many countries the ratio between the number of old and young people is constantly growing. Figure 1, drawn from data on the Italian population, shows that the percentage of people 'over 80' relative to the total population might increase eight-fold from 1950 to 2030. This means that the number of old people will be a sizeable percentage of the whole population. We want these people to have a very good quality of life while keeping expenses as low as possible. In many cases this means helping people stay home as long as possible (no more than 5% of the elderly in Europe are institutionalized).

The system we are building is targeted at raising as much as possible the age at which a person needs to be institutionalized. We believe that constant monitoring through pervasive technologies is essential to provide more efficient health assistance at home. In fact, recent studies on the acceptance of technologies for the elderly [1, 2] show that while people tend to look for social relationships in activities such as cleaning or playing cards, in situations related to safety, health and personal care they are also likely to rely on technological solutions.

For this reason there has been a strong development of computer technologies applied to specific fields of medical sciences in order to allow the delivery of clinical care outside of hospitals. For example, telemedicine and clinical decision support systems have been used to collect complex clinical data and implement diagnosis at-a-distance. The practical use of these techniques in real contexts has shown that they work well for some very specific healthcare applications, such as medical prescriptions [3] or real-time transmission of clinical data. Our system complements these techniques by taking into account the contextual setting and the health evolution of patients over long periods of time.

In our study we address those elderly who are clinically stable although they might be affected by chronic diseases and physical decline (more than 90% of the population over 65 has more than one chronic disease). Since their health conditions do not require constant monitoring of complex biomedical parameters, these patients do not need, and are less tolerant of, invasive sensors.

Many user-centred systems that analyse user's behaviour and detect emergencies have been developed. They often cater to the identification of predefined patterns of behaviour rather than to

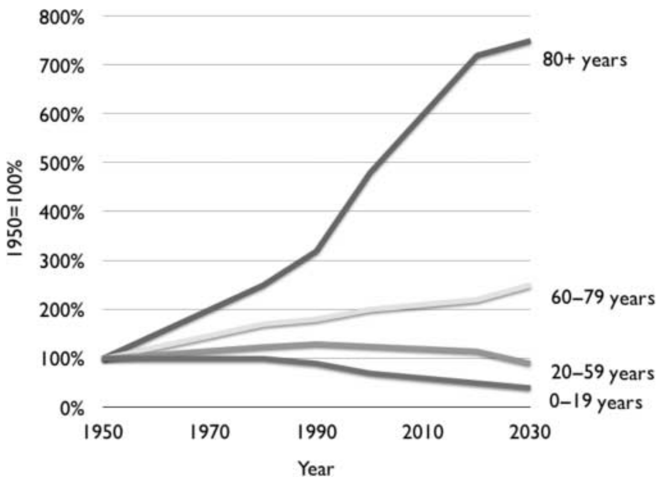


**FIGURE 1.** Size of the Italian population by age groups: each age group is separately plotted and normalized to the group size in 1950 (source ISTAT and IRP-CNR).

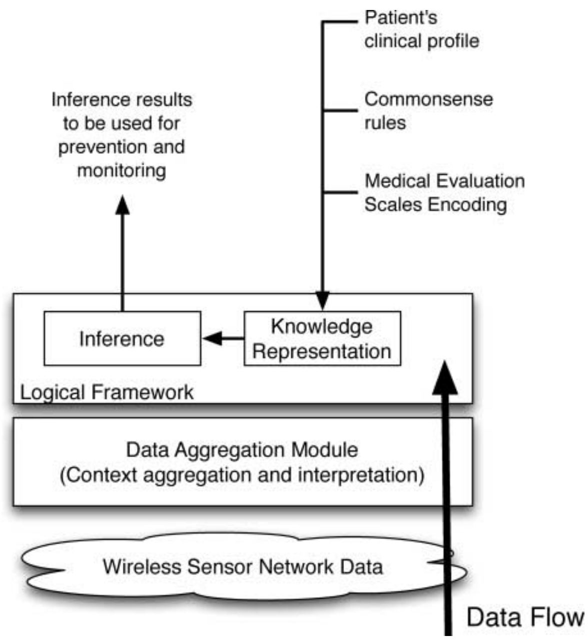


**FIGURE 2.** Data flow.

the assessment of health in general, and they are mainly based on statistical analysis of data, thus needing substantial training to be adapted to a particular patient.

We use monitoring to support prevention, causal diagnosis and emergency detection in the same framework and provide a global representation and reasoning model for general health assessment, combining medical knowledge, patient's clinical profile and context evaluation through sensor data. These data are combined and interpreted by an inference engine to help caregivers detect patients' physical, mental and social status as it evolves.

Figure 2 shows a very high-level overview of the architecture of our system and the correlation between its components.

The presence of heterogeneous information makes it possible to both automatically adapt the results of the reasoning process when new information is available and deal with user and context-specific constraints. We use answer set programming (ASP) because it constitutes a powerful declarative framework for knowledge representation and reasoning in this application context. ASP addresses many of the requirements listed in Section 1.1.

We do not focus on the use of robots for healthcare because, beyond their high cost of set-up and maintenance, their presence is rather intrusive and the help they can really provide is marginal.

Starting from these considerations, we have developed the first prototype of the SINDI (Secure and INDependent LIving) system. The principal requirements of the system and the way we address them are illustrated in Section 1.1.

### 1.1. System requirements and design issues

As already mentioned in the introduction, the SINDI system has been designed to support caregivers in monitoring and providing health assistance to the elderly in their home environment by using wireless sensor technologies and automated reasoning capabilities, but it also interacts with the elderly to help them directly. The main aim is not to extend life but to enhance autonomy and increase quality of life.

For this reason, the main requirements we have been focusing on are:

- *unobtrusiveness*: the monitoring system should not affect the lifestyle and habits of the person being monitored;
- *technological soundness*: the monitoring system should use what is already commercially available with respect to technology;
- *affordability*: costs should be kept low in order to be affordable by medium-income families; set-up and management should also be easy and cheap;
- *user-friendly*: the elderly may have problems in handling complex multiple devices, therefore the interface with the system should be as close as possible to what they are used to, and the interaction should be intuitive;
- *medical soundness*: though SINDI does not support complex medical diagnosis of specific diseases, the intelligent support should take into account the appropriate medical knowledge;
- *user-centrality*: each person has different needs, thus a system that is supposed to work in a specific home environment should consider the psycho-social and clinical setting of the patient when evaluating the evolution of his health status, rather than mapping his situation to similar medical cases;

- *adaptivity*: to address user-centrality, the system should incorporate mechanisms to adapt to different patients, both automatically and by explicit parameters setting;
- *context-awareness*: sensing activity and reasoning support should consider not only static user-specific needs, but also the evolving state of the patient and of the environment in order to give more accurate results when data are incomplete and dynamic;
- *reactivity*: for a long-term monitoring support, data manipulation and interpretation can be done offline; however, the system should be able to react in real time to specific triggers (such as emergency situations, user input, system feedbacks);
- *reliability*: collection and aggregation of data, as well as results of reasoning used to help caregivers in understanding patient's health evolution, should be reliable enough to assure adequate support;
- *accuracy*: results of the reasoning process should be as close as possible to what caregivers expect, according to the available information.

In order to address these requirements, some technical and methodological choices turned out to be crucial in the design and implementation of the SINDI system:

- To preserve unobtrusiveness, we decided not to use cameras in order to avoid the uncomfortable feeling of being constantly observed. Dynamic data about the person and the environment are unobtrusively captured by a wireless sensor network (WSN), composed of several sensor nodes, a wearable monitoring device and a master processor.
- The use of commercial nodes readily available on the WSN market considerably increases the affordability of this kind of system, at the same time helping the reduction of overall costs and simplifying configuration and deployment.
- Although the elder generation is getting closer and closer to technology, they might have problems in dealing with complex devices such as a PDA or devices that force them to read from a small screen, such as portable phones. For this reason, SINDI allows interactions with the patient through the TV screen, controlled by a device that is similar to a TV remote.
- Context-awareness is another important requirement. The reasoning has its basis on the aggregation and interpretation of different kinds of information from heterogeneous sources (such as light, position, movement, localization, load cells). The idea is that additional information can help in characterizing the solutions of the reasoning process, identifying the most plausible ones, according to the available domain knowledge. This also enhances reliability since heterogeneous sources of information that can be interpreted may help in compensating errors and incompleteness of data.
- If and when new sensors are available, the information they produce can be easily taken into account by adding new rules.
- The need to make the system user-centred and medically sound leads us to include some medical knowledge in the reasoning phase. In this way it is possible to trace general habits and their correlation with the patient's well-being according to the evaluation methods of clinical practice. In particular, we want to address the fact that clinicians need to be supported in:

  (i) understanding patients' physical, mental and social settings as they evolve,
  (ii) predicting what could follow with respect to particular changes in one or more aspects of patients' general health state and
  (iii) identifying correlated aspects that may be the cause for a negative change in the patient's general health state.

The first aspect is related to the contextualization of worsenings of the general health status of the patient, not only with respect to similar clinical cases, but giving more importance to aspects that turn out to be important for the patient at hand. The second aspect refers to prediction, i.e. the identification of health-related aspects (we refer to them as *items*) that deserve specific attention regarding a worsening; this makes it possible to act *before* more serious side-effects are observed and to plan appropriate short- and long-term interventions, thus reducing risks. The third aspect is more similar to diagnosis, but it should be a *local* process rather than a case-based one, in that it must take into account patient's clinical and environmental settings and adapt to the specific patient.

- To perform these reasoning tasks and encode the relative knowledge into a common model, we believe ASP is the right framework because:

  (i) the effectiveness of the implementation makes it possible to express deductions, default reasoning, constraints, choices and qualitative preferences;
  (ii) declarativity allows the automatic encoding of medical knowledge, thus making the system easily extensible and medically sound;
  (iii) the use of contextual information and the way new knowledge can be taken into account, makes it possible to deal with incomplete information and enhance context-awareness.

- Medical soundness and context-awareness can help obtain better solutions because the combination of different sources of information (sensors, medical knowledge, clinical profile, user-defined constraints) that change over time make the system more reliable (i.e. much better able to disambiguate situations, thus reducing false positives) and adaptable (easily extended on the face of new available information).

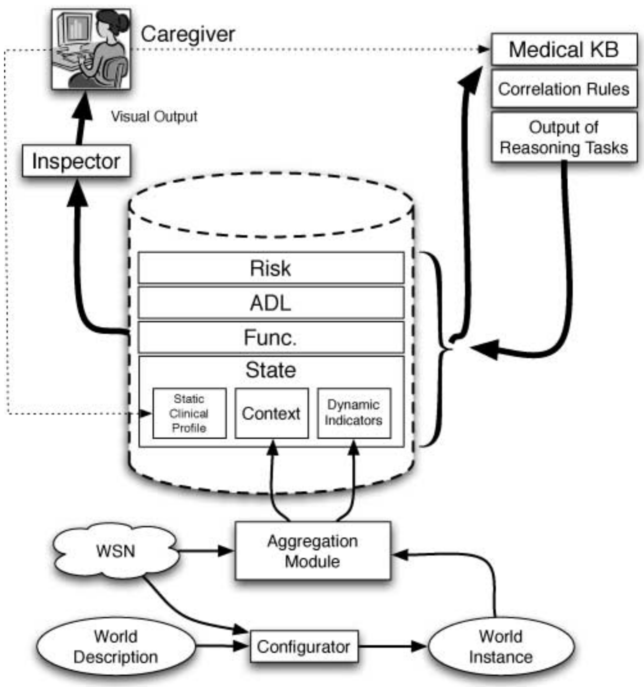


**FIGURE 3.** Architecture overview.

- We also considered the fact that the reasoning process is run not only periodically but also according to specific triggers. These triggers can be associated with states of emergency and specific actions to be performed by the system. By adding appropriate logical constraints to the ASP program, emergencies can also be contextualized in almost real time. This behaviour addresses reactivity.

The general architecture of SINDI and its main components are illustrated in Fig. 3.

Section 2 describes the design and implementation of the WSNs supporting SINDI's intelligence. The reasoning framework is formally described in Section 3, while details about the implementation of the knowledge representation and reasoning model are presented in Section 4. Section 5 reports some preliminary evaluations and Section 6 contains concluding remarks.

## 2. INTELLIGENT SENSOR NETWORKS FOR HEALTHCARE: OUR VIEW

Wireless Sensor Networks [4, 5] consist of nodes that are capable of interacting with the environment by sensing or controlling physical parameters. These nodes use packet radio communication to collaborate in order to complete their tasks. This kind of network is typically used to collect data for long periods of time without assistance. Specific scenarios for WSNs include habitat monitoring, industrial control, embedded sensing, medical data collection, building automation, fire detection, traffic monitoring.

In the last few years, many interesting systems were developed in the area of WSNs for assisted living and healthcare, among which we mention ALARM-NET [6], Sensor Assisted Independent Living Networks (SAILNet) [7] and CodeBlue [8].

ALARM-NET is a WSN designed for long-term health monitoring in assisted living and residential environments. The central design aim was to adapt the behaviour of the system, including power management and privacy policy enforcement, to the individual life patterns that are analysed and fed into the system. The system incorporates a circadian activity rhythm (CAR) analysis module used in all the reasoning about the activities performed by the users. SAILNet proposes to apply the technology of WSNs as a nonobtrusive tool to monitor the activities of elders living in their apartments, focusing only on fall detection and pointing out that quick responses to these alarms are the critical requirement. Therefore, the project gives much emphasis to the availability of WSNs. CodeBlue is a wireless communications infrastructure for critical care environments. It is designed to provide routing, naming, discovery and security for wireless medical sensors, PDAs, PCs and other devices that may be used to monitor and treat patients in a range of medical settings. Given our application scenario, the cited projects do not fulfil all requirements.

Similarly to the systems described above, the WSN of SINDI monitors environmental data, physical behaviour and weight of individuals. In our framework, in order to fulfil the requirements of SINDI's monitoring, we need to manage specific aspects of WSNs such as (i) hierarchical organization and topology control, (ii) positioning, localization and tracking, (iii) synchronization and power management and (iv) deployment and network configuration.

The WSN we use in SINDI collects environmental data on light, temperature, humidity and opening and closing of doors and windows. Data on user's movement (see Section 2.3), localization and weight (using load cells under a bed or an armchair) are also collected. Our aim is to acquire all possible information about the context in which the user lives instead of focusing only on medical information.

### 2.1. WSN architecture

In real deployments, our assistive monitoring system must be able to cope with dozens of nodes for every room/area. Moreover, we should consider that the number of nodes can grow up to one hundred for large environments. Tracking the position and monitoring the movements of a user could generate a large amount of network traffic.

Therefore, we need a nontrivial network organization to manage the WSN of SINDI. In our view, an hierarchical network organization helps to solve these issues.

The architecture of SINDI's WSN is composed of:

- a base node in every zone, always active and connected to household power, mostly used for network coordination but also with its own sensing capabilities;
- environment nodes (battery powered) for sensing the environment data or capturing particular events;
- a wearable monitoring device;
- a master processor.

Both the base nodes and the environmental nodes can sense at least: temperature (10–40° with 1° precision), humidity (low–normal–high), light changes that are meaningful to people, and received signal strength indication (RSSI) for localization and proximity. Furthermore, the base node integrates a power supply (it looks like a telephone charger), a small rechargeable battery for power outages and can sense the presence of AC supply. The wearable monitoring device is used for the user's localization and includes several sensors (accelerometers, gyroscopes and magnetometers) for movement detection. The master processor is the coordinator node of the network. It is the gateway of the network and it has storage, processing power and main memory capabilities in the ballpark of an average PC.

The network is organized hierarchically. The environment in which the user lives is divided into zones and every zone is controlled by one base node. Moreover, every zone can be divided into several sensing areas where one or more environment nodes operate. The wearable node can move from zone to zone without loss of connectivity. The master processor manages the entire network applying topology-control mechanisms and routing algorithms. A graphical representation of the network organization is also shown in Fig. 4.

In the past few years, many new topology-control and data-routing algorithms have been proposed for hierarchical network organization in WSNs.

Topology control consists in deliberately restricting the set of neighbouring nodes of a given node in order to minimize network traffic and maximize node power. An example of hierarchical topology control is described in [9].

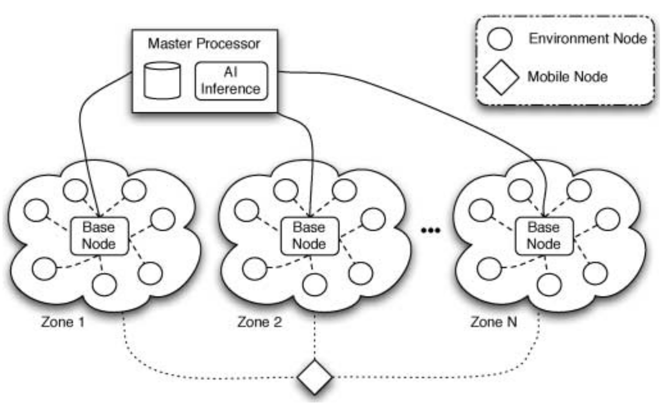


**FIGURE 4.** Wireless sensor network architecture.

Routing mechanisms take into account the characteristics of sensor nodes along with the application and architecture requirements. Almost all WSNs routing protocols can be classified as either data-centric, hierarchical or location based although there are a few protocols based on network flow or quality of service awareness.

Approaches to hierarchical routing like those described in Low-Energy Adaptive clustering Hierarchy [10], Threshold sensitive Energy Efficient sensor Network [11] or Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network [12] are particularly interesting for their approach to node organization in clusters (in our settings, a cluster is a zone of the environment).

The medium allocation (MAC) is handled with the industry standard IEEE 802.15.4 protocol.

Our hierarchical network organization has several advantages. First of all, the reactivity of the network to unexpected events is significantly increased by the presence of a base node always active in every zone of the environment and therefore quick responses to alarm situations are possible. Moreover, the localization and tracking of the user at zone level is always possible (obviously, a better accuracy can be reached using the environmental nodes). The environmental nodes also have lower power consumption because the network traffic is reduced by turning to sleep all the areas where the patient is not detected. Finally, the process of configuration and the deployment of the network is notably easier with this organization.

In the following subsections we give more details about these aspects.

### 2.2. WSN organization

As we have seen, the WSN of SINDI collects several types of user and environmental data and every node of the network has various sensing capabilities. The network nodes are all similar, but the on-board sensors are heterogeneous. In our system, we need to configure the behaviour of every node, changing dynamically its sensing capabilities. Therefore, we need to use specific software to correctly configure and manage the network.

Several middleware environments that provide routing, data aggregation and communication services have been designed in order to optimize and limit resource consumption. For instance, middleware like [13] or [14] are good examples of generic environments based on message exchange mechanisms optimized for homogeneous sensor networks. These environments, however, do not provide enough flexibility to manage heterogeneous sensors and do not meet all our requirements.

We developed a middleware that provides the following functionalities: (i) dynamic configuration of the nodes and simple data aggregation; (ii) communication, routing, power control and synchronization; (iii) positioning, localization and tracking.

#### 2.2.1. Dynamical configuration of the nodes

Every node of the network has several *sensory capabilities*. A base node (or the master processor node) manages the capabilities of the environmental nodes, enabling or disabling them when needed. In our settings, these capabilities are represented with a bit mask, where every bit represents a sensory capability (such as bit 0 for temperature, bit 1 for humidity, bit 2 for environmental light, and so on.). Therefore, the configuration of the environment nodes can be easily managed and single activities can be turned on or off by a base node in every zone or by the master processor node.

In order to reduce the traffic in the network and eliminate redundant data, we developed a middleware that manages simple aggregation rules [15] at the zone level, e.g. temperature is aggregated in each zone unless the person is in the area and the temperature at specific locations is required.

Moreover, the zone-based network configuration we use in our network requires specific configuration tools. These tools allow the association of nodes with rooms and areas in a very simple fashion. Due to the presence of specific and heterogeneous sensors the current version of the system does not permit the use of fully automatic configuration tools, therefore we need a scene-analysis phase in which we configure the system for every specific environment.

#### 2.2.2. Communication, routing, power control and synchronization

The flow of messages follows the hierarchical structure of the network. Depending on its active capabilities and within the same awake cycle, an environmental node reads its sensor values and sends a message to its base node. In our test setting, the value of the battery level is always transmitted in order to monitor the network consumption. An environmental node can easily change its behaviour when it receives a new configuration message from a base node. After receiving all messages or after a small timeout, every base node sends the data to the master processor for storage and further elaboration.

Finally, the presence of a base node always active in every zone simplifies data routing because the environmental nodes are guaranteed to always find a listening base node. Therefore, they could simply send a message with the proper sensory data and quickly enter sleep mode without wasting precious energy.

Most of WSN research is focused on the optimization of node resources. In order to make the batteries last as long as possible, the duty cycle of a node should be more or less 1%. Moreover, configuring all the environmental nodes to always use all their capabilities for every active cycle is wasteful.

A significant reduction of energy consumption could be achieved by increasing the sleep time of the environmental nodes when possible. For example, one might set the environmental nodes to sleep for very long cycles during the night. Consequently, most of the resources saved at night could be used in other moments of the day.

#### 2.2.3. Positioning, localization and tracking

Given our application scenario, we do not need a high-precision localization and tracking system. Real-time, high-precision tracking of the user does not give our system significant advantages. Therefore, a range-free algorithm that uses proximity-based techniques turns out to be sufficient to accomplish our needs. Instead of computing the precise spatial position of the patient we divide the environment into several *logical* locations, using them for the tracking algorithm.

However, higher accuracy could be achieved for particular events or alarms by using all the nodes in a zone at the same time. The locations and tracking computations are distributed among the base nodes in order to the minimize energy consumption of the environmental nodes.

The wearable monitoring device broadcasts a localization message every 2 s in order to provide an RSSI-based position estimation to the other network nodes. These messages have a time to live (TTL) equal to 1 to guarantee that only the nodes closer to the monitoring device receive them without further routing. This is necessary in order to retrieve coherent values of the RSSI. We used simple scene-analysis techniques to optimize the localization and tracking system. Furthermore, the resource consumption of the wearable device is considerably reduced by using the on-board inertial sensors. When the user is stationary, we limit the message broadcast using threshold detection algorithms.

Section 5 presents an evaluation of the performance of the tracking component.

### 2.3. Activity recognition using wearable sensors

Detecting and analysing patient's movements are key factors in our home healthcare solution. A viable solution to understanding patient's activities is to use inertial wearable sensors [16–18]. An inertial sensor is well accepted because it is small enough to go unnoticed when worn.

To understand movements, we use a data-gathering device with a triaxial accelerometer, gyroscope and magnetometer on board. We use only one device worn on the hip of the patient, oriented as follows. The sensor is positioned on the right side of the pelvis; the $y$-axis of the sensor reference system is directed to the ground, parallel to the axis of the body; the $x$-axis is directed to the back of the person; the $z$-axis is parallel to the pelvis axis and directed outside the body (see Fig. 5).

We assume that the device is always correctly positioned with the same orientation. The shape of the device helps the patient to correctly wear the sensor. Using the device on the hip, we are able to acquire the data generated by the movement of the centre of the mass of the body. Peripheral body movements (such as hands, arms movements, the foot impact angle, the head inclination respect to the body) cannot be detected because

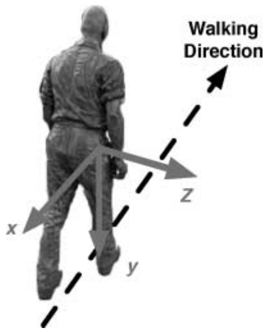


**FIGURE 5.** Mobile node positioning.

they cannot be sensed at hip level. Using only one device is a limitation to what we can understand about the patient's activities [19, 20], but the smaller computational cost and the stronger social acceptance of having only one device to be worn are strong advantages.

### 2.3.1. *Understanding movements*

The activities we need to understand are of three types. One relative to the way the body is set in space: lying, sitting, standing. Another, relative to the movement of a person through space: walking, stepping upstairs/downstairs, staying. The last, relative to the changes of direction when the person is moving: forward, right/left turns, back turns.

Data are usually sampled on the wearable device at 50 Hz, they flow through the WSN and are analysed on the server side at runtime. Some critical events, like falls, can happen and we need to recognize them very quickly, but we must acquire enough data to compute a number of features sufficient for the understanding algorithms. We store the data generated by the accelerometers in a temporary buffer (acquisition window) that contains 10 s of acquired data. This dimension has proved to be large enough to recognize our set of activities, but small enough to guarantee that a critical event can be quickly recognized. The acquisition window must be refreshed and filled with new data as they are produced by the accelerometers. In order to identify events or actions that happen on the borders of our acquisition window, the new acquisition window is overlapped with the old one by a half of its dimension. Recognized activities are tagged with initial and end time and are saved in a SQL database, where they will be used by the reasoning module for immediate or later consideration.

### 2.3.2. *Segmentation*

The data we acquire are a set of ordered values in time. Analysed in separate dimensions, they appear as a one-dimensional function in the time domain. First of all, we must correctly segment the signal, as the quality of the features used for activity recognition is very sensible to segmentation accuracy. For this analysis, we use the module of accelerations, which is a quantity independent from sensor orientation.

$$\text{magn\_accel}(t) = \sqrt{x_t^2 + y_t^2 + z_t^2}. \qquad (1)$$

We are interested in identifying a well-known set of predefined actions (see Table 4 in Section 5). The dispersion of the signal around the average has proved to be a particularly useful feature for our aim.

We consider the signal as a population of samples varying around an unknown average: we calculate the average and dispersion with a moving window of size $2k+1$. The dispersion of the signal is a highly user dependent value, can be subjective, and depends on many particular factors. For a given action this value can be different even when the same person repeats it, for example, with more force, or determination, or velocity; but when a person changes movement, e.g. turning left or right during a walking activity, the variance of the signal changes significantly. Differences in the variance function are points in time useful to identify when an action starts or stops.

We use mean error (ME) that is related to the variance of the signal. Differently from the variance, ME does not use power operators, reducing computational costs. The dimension of the moving window is $k = 25$ and has been chosen heuristically. This algorithm is also used to understand when the person is still. A person that is not moving generates a signal with a very low variance; we consider a person still when ME $< 1$.

### 2.3.3. *Walking*

Walking is recognized by using a decision tree (DT) algorithm. Walking is a movement that roughly appears as a sequence of regularly spaced signal peaks: higher peaks coincide with the right and left steps (see Fig. 6). We detect peaks using a peak detection algorithm, then we try to understand if they are equally spaced in time by simply measuring the mutual distance of peaks. If they are equal with a tolerance of 2% we consider the movement 'regular'. The signal generated during

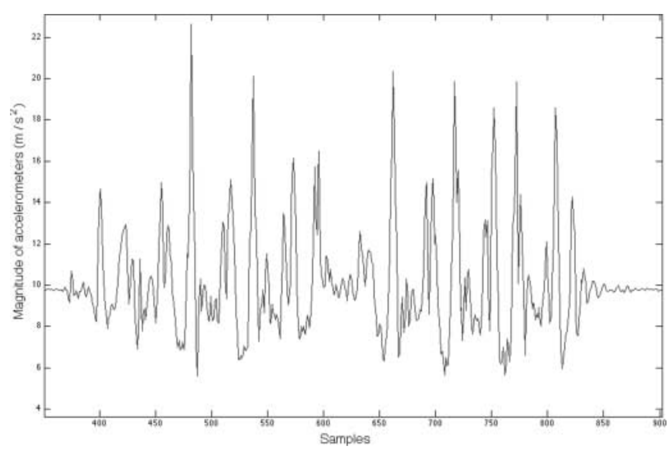


**FIGURE 6.** An example of walking (a person makes five steps forward, turns, and then make five steps backwards).

movements is quite complex and rich in personal features, so we must denoise it before using the peak detection algorithm, to eliminate many 'false peaks' that are normally present. We use a FFT with a 15 Hz low pass filter: this value helps to maintain the most important information of the signal, eliminating all the noisy details. Parameters of this algorithm have been chosen heuristically.

#### 2.3.4. Direction

To understand direction we use the information coming from the $x$- and $z$-axes of the magnetometer. The $\tan^{-1}(x_t/z_t)$ returns the angle of the body with respect to the north pole. This operation is quite simple, but in this scenario we must solve two problems. First, we have to distinguish minor movements from actual changes of direction. During walking activities the hip rotates around the ideal centre of the pelvis creating a sinusoidal-like signal that is not related to real changes of direction. The second problem is to distinguish low rotations of the body from an actual and sudden change of direction. For example, a person making a large 360° circular trajectory is not making what is usually considered a '360° turn'. This makes a difference at the reasoning level where a turn is considered a sudden movement and not a slow and unintentional action. This is an important semantic difference that must be taken into account. Given $\tan^{-1}(x_t/z_t)$ we smooth this function to eliminate artefacts due to the pelvis movements, and we use the derivative of the function to understand when a sudden change of direction occurs. The quicker the movement, the higher the derivative. When the derivative increases to more than a specific threshold we consider the movement a significative turn, and we track the zero crossing prior and after this point to understand where the turn really started. The angle difference and the sign tell us about the magnitude of the turn in degrees and if the turn is in the right or left direction.

#### 2.3.5. Position of the body in space

For the position of the body in space we look at how the acceleration vector **g** is positioned with respect to the $y$-axis of the sensor. A force is always present on earth, the gravitational force: accelerometers can measure the intensity of the **g** acceleration correlated to this force. When the acceleration vector **g** is parallel to the $y$-axis, the body is parallel to it too, and the $y$ value of the accelerometer tends to be near the **g** value, as its projection is maximum on $y$. When the body is lying horizontally, the projection of vector **g** is orthogonal to the $y$-axis and tends to be zero, on average. Sitting is an intermediate position, with an intermediate value. We diminish the effect of outliers using a mobile smooth operator (window of $2k + 1$ dimension, $k = 40$). After smoothing we use a threshold to decide the position of the body in space. We consider the value $y > 7.7$ as standing, $y < 3.2$ as lying and $3.2 < y < 7.7$ as sitting.

#### 2.3.6. Falls

Falls are considered critical events for an elder person, and must be avoided. But if a fall occurs, we would like to detect it in order to start an emergency process. When a body falls, it tends to follow an inertial trajectory and the acceleration vector **g** tends to became null in the sensors reference system. Then, as a consequence of the impact, we register very high accelerations and rotation values. Therefore, a fall is detected using a DT algorithm. First, we check if the magnitude of acceleration becomes smaller than $5\,\mathrm{m/s^2}$. If this is the case we control the rate of turn, the acceleration magnitude, and its derivative values.

## 3. THE LOGICAL FRAMEWORK

The declarative logical framework we use is that of ASP. ASP is based on the *stable model* semantics proposed by Gelfond and Lifschitz [21]. The stable model semantics is a declarative semantics for logic programs with negation that can be seen as bringing together concepts and results from logic programming and default reasoning (see Section 3.1 for details).

Thanks to its expressiveness and to the availability of efficient implementations [22–25], ASP has started to play a relevant role in solving complex knowledge representation and reasoning problems [26, 27].

Some preliminary notions are needed in order to understand the strength of the ASP formalism.

### 3.1. ASP: some notions

In ASP a given problem is represented by a logic program viewed as a description of properties and constraints on the solution. The logic program is written in such a way that results of the evaluation of the program correspond to the solutions of the original computational problem. These results are given in terms of *answer sets*.

An *answer set* for a logic program is a minimal (in the sense of set-inclusion) set of literals satisfying the logic rules included in the program.

A rule $r$ is an expression of the form

$$R_j : L_0 \leftarrow L_1, \ldots, L_m, \ \textit{not} \ \ L_{m+1}, \ldots, \ \textit{not} \ \ L_n, \quad (2)$$

where $L_i (i = 0 \ldots n)$ are literals, *not* is a logical connective called *negation as failure* and $n \geq m \geq 0$. We define $L_0 = \text{head}(R_j)$ as the *head* of rule $R_j$, and $\text{body}(R_j) = L_1, \ldots, L_m, \textit{not} \ L_{m+1}, \ldots, \textit{not} \ L_n$ as the *body* of $R_j$.

Furthermore, let $\text{body}^+(R_j) = \{L_1, \ldots, L_m\}$ and $\text{body}^-(R_j) = \{\textit{not} \ L_{m+1}, \ldots, \textit{not} \ L_n\}$.

Rules $R_j$ with $\text{head}(R_j) = \emptyset$ are called *integrity constraints*, while if $\text{body}(R_j) = \emptyset$, we refer to $R_j$ as *a fact*. Rules with variables are taken as a shorthand for the sets of all their ground instantiations.

In answer set programs there are no function symbols and the role of recursion is dramatically restricted, thus enabling a very efficient computation of the solution. Another appealing feature of ASP is the possibility of assuming that something is true and later retracting this conclusion when new knowledge is available. We refer to this capability as Default Reasoning. In ASP one can express default reasoning by specifying general knowledge for standard cases (the defaults) and using negation as failure to introduce possible exceptions. As long as there is no evidence of the truth of an exception, the default holds. New rules can be modularly added to infer the truth of exceptions (see Example 1). Default reasoning is said to be *nonmonotonic* in that, adding new knowledge, you may be no longer able to conclude what you could before. Such reasoning capability makes it possible to deal with incomplete knowledge.

The power of the ASP formalism is also due to its close connections to the field of satisfiability checking and constraint satisfaction, declarativity and expressive power allowing the modelling of nondeterministic choices, priorities and cardinality constraints with compact encodings.

Thanks to the efficient implementations available, nondeterministic choices and constraints can be used to *generate* all possible solutions for a problem and *test* them via constraint satisfaction (see Example 2).

These features of ASP make it a powerful knowledge representation and reasoning framework dealing with abductive reasoning (finding consistent explanations of some given observations), belief revision, decision problem solving, planning and diagnosis.

### 3.2. Knowledge representation of the home healthcare domain

A careful analysis of health care in home settings suggests that health-related items can be classified into three levels: functionality level representing functional disabilities of the person monitored, activities of daily living (ADL) level representing their dependence in performing daily activities, and risk assessment level characterizing risky conditions. Significant aspects of health assessment at each level (referred to as *items*) have been identified according to the medical practice in health assessment of the elderly [28] and encoded in our declarative framework (see Sections 3.2.2–3.2.4 for details).

A lower layer (state level) contains aggregated context data as well as static and dynamic evaluations of significant aspects of patient's clinical settings (referred to as *indicators*).

Static aspects of clinical profiles include stable pathologies characterizing elderly care and drug intake as well as results of specific complex tests performed periodically by the caregiver. Predicates used in logic rules to represent the static profile are illustrated in Table 1.

**TABLE 1.** Logical encoding of patient's profile.

| Predicate | Description |
|---|---|
| test(Name, Value) | Test results |
| drug(Name) | List of drugs |
| pathol(Name) | List of pathologies |
| profile($X$, Name, $V$) | $X = \{$drug, pathol$\}$ |
| | $V = \{$yes, no$\}$ |

**TABLE 2.** Logical encoding of dynamic profile.

| Predicate | Description |
|---|---|
| lev($L$, $I$) | Association items-level |
| obs($I$, $V_i$, $T1$, $T2$) | Evaluation of an item |
| obsind(Ind, $V_i$, $T1$, $T2$) | Evaluation of an indicator |
| link($I$, Ind) | Association item-indicator |
| range(Ind, $V_i$) | Range of values |
| ord(Ind, $V_i$, Num) | Order of values |

Dynamic aspects of the clinical profile include:

- indicators that can be evaluated through ad-hoc tests proposed by the system when needed;
- indicators evaluated by the WSN through continuous monitoring and data aggregation;
- indicators evaluated through logic rules (e.g. the quality of sleep).

Every time the inference process is run, the system compares values of indicators from the previous inference with the actual values (either aggregated by the sensors or inferred by logic rules) and computes differential evaluations. Admissible values for each indicator are part of the medical knowledge and are encoded by the knowledge engineer.

At higher levels, each indicator can be associated with one or more items. Results of differential evaluations identify which indicators are subject to worsenings and which items are critical: the higher the number of worsenings associated with an item, the more critical the item is.

Predicates used in logic rules for evaluation are listed in Table 2, where $I$ represents an item, $L$ is a level, $V_i$ are values, Ind represents indicators and $T1$, $T2$ are timestamps. Logic rules used to detect worsenings by using differential evaluations are presented in Section 4.

We wish to point out that evaluations of items obtained by the system through specific tests are only partial. In fact, computer-aided tests can include only a few of the aspects considered in the complete initial evaluation performed by the caregivers. The frequency with which these tests are proposed to the user depends on test scheduling done by the caregiver. As an example, if the quality of sleep becomes substantially worse, this can raise a trigger and the system proposes a computer-aided version of the appropriate test; the worsening can also be observed by the caregiver who remotely schedules the test for the patient.

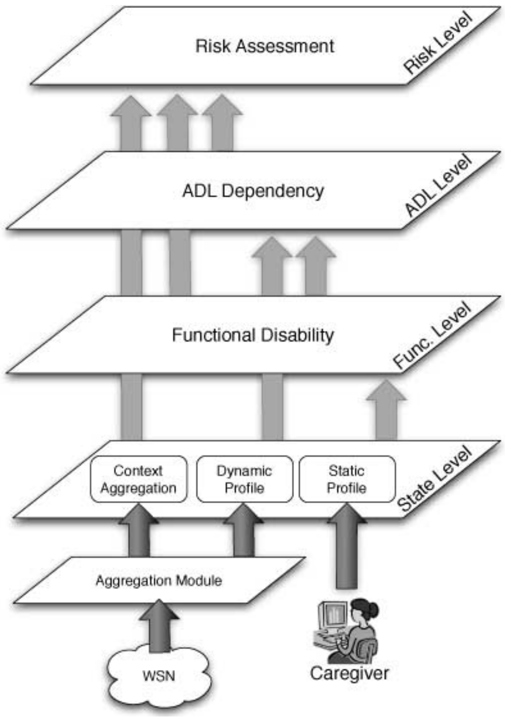


**FIGURE 7.** Information flow across levels.

The reasoning process takes also into account medical knowledge about causal correlations among items, and combines them with results of differential evaluations to show how the patient's health can evolve in terms of functional disability (functionality level), dependencies in performing daily activities (ADL level) and risks assessment (risk assessment level).

Figure 7 illustrates the flow of information across levels: details about the person and the environment at the state level are provided by the WSN (eventually aggregated), while values of items at upper levels are hierarchically influenced by values of items at lower levels.

In the following subsections we give details about items at each level and indicators (at the state level) associated with them. Details about causal correlations and reasoning capabilities are presented in Section 4.

### *3.2.1. State level*

The *state level* includes static and dynamic profiles as well as context-dependent information, which are the results of the aggregation of sensor data. The static profile includes:

- complete test results evaluated by the caregiver on periodic examinations using appropriate medical scales: cognitive state (mini mental test [29] and clock drawing test [30]), vision (optical tests), mobility (Tinetti-Performance Oriented Mobility Assessment (POMA) scale [31]), affective state (Geriatric Depression Scale (GDS) test [32], nutritional state (mini nutritional assessment [33]) and ADL dependency (Katz scale [34]);
- intake of drugs, among which we consider ache inhibitors, benzodiazepines, psychotropes, neurolectic and anti-parkinson;
- presence of specific age-related pathologies, among which we consider reduction in visual acuity, hearing loss, osteoarthritis, cognitive decline, depression, alcoholism, vascular pathologies, arthritis, cardiac problems, parkinson, epilepsy, dentistry problems, disthyroidism, acute pathologies;

Interesting pathologies and drugs, as well as valid tests, have been identified together with the geriatrics of the S. Gerardo Hospital in Monza. The declarative nature of the ASP framework makes it easy to add new information. However, we are aware of the fact that a user-friendly interface should be available for clinicians to extend the medical knowledge of the system without the constant support of a knowledge engineer. We believe this is feasible since declarativity allows automatic encoding from a high-level specification into ASP, through appropriate mapping. We are investigating this issue.

Context-dependent information includes:

- *Personal details:* biomedical parameters such as temperature and weight;
- *Environmental properties:* average light value, humidity, temperature, architectural barriers (such as presence of stairs or carpets in a given room or area of interest);
- *Basic activities:* movement activity can be easily captured by the wearable sensor, and we consider it as characterized by *motion* (walking, standing still), *position* (sit, lay, stand) and *orientation* (straight, turning);
- *Localization:* the way the patient moves from one room/area to another is traced;
- *Interaction with objects:* we consider two kinds of binary interaction according to sensors associated with the specific object, i.e. *pressure* (chair and bed objects) and *switch* (doors, windows and devices).

Context-related data can also be used to infer values of those indicators that are not directly available from aggregated sensor data. As an example, consider the indicator *quality of sleep*. To understand the quality of sleep it is necessary to reason about the night activity, taking into account consistent contextual information (location, state of objects, movement) and some auxiliary predicates (such as start/end of night time, getting up, going out of bed). A simplified version of the ASP code to infer quality of sleep is shown in Section 4.

Consistent interpretations of the context can also be crucial for caregivers in order to investigate the particular settings in which worsenings or emergencies are detected, since these values can be analysed through a visual interface similar to the one illustrated in Fig. 8.

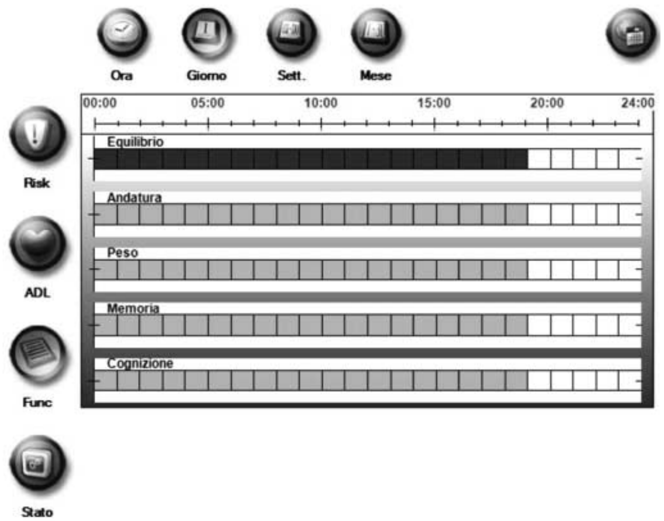


**FIGURE 8.** Caregiver interface to access to context data.

Items at each upper level are characterized by an initial static evaluation (predicate obs() in Table 2), and a set of indicators used for differential evaluation (predicate obsind() in Table 2). As an example, a complete test performed by the caregiver is encoded by an obs() predicate, while partial (computer-aided) versions of the same test are encoded as indicators using obsind() predicates.

In the following subsection we identify preliminary tests used for initial evaluation and give additional details about the association of indicators with items.

### 3.2.2. *Functionality level*

At this level, the system considers the following functional disabilities:

- balance and gait, initially evaluated through the appropriate parts of the Tinetti-POMA medical scale; indicators are represented by aspects of the scale that can be captured and evaluated through the wearable sensor, i.e. standing, sitting, turning and walking;
- nutrition, initially evaluated with the mini nutritional assessment test; the indicator is the body mass index (BMI);
- vision, initially evaluated through specific optometric tests; indicators are the level of light during the day and at the sunset (according to medical practice, keeping the light on when the blinds are opened and the natural light coming from the outside is up to a certain level, may indicate a problem);
- hearing sensibility, evaluated through audiometric tests; the indicator is the reaction time to a ringing bell;
- mental and cognitive capabilities, evaluated through the mini mental and clock drawing tests; indicators are represented by a computer-aided questionnaire, counting ability and quality of sleep;
- insomnia, evaluated through a questionnaire; indicator is the quality of sleep;
- emotional stability, initially evaluated through the GDS test [32]; indicator is the computer-aided version of the GDS test.

According to the literature, mobility remains one of the most important aspects to be assessed in order to protect the elderly from the negative consequences of a fall. The Tinetti-POMA test has been claimed to be the gold standard in assessing mobility dysfunctions in the elderly and is an important fall risk assessment measure. This test was published by Tinetti in 1986 and it has been designed to evaluate the position changes and gait manoeuvres used during normal daily activities.

With respect to complete tests and then computer-aided versions, a concrete example is represented by the mini mental and clock drawing tests. These tests are not easily performed without someone's assistance. Thus, the evaluation of the correspondent indicators is done by periodically (once a month) proposing a reduced version of the test to the user via TV screen: the abbreviated mental test (AMT) consisting of a small set of questions [35].

Results of these simple tests are stored in the DB and they can be accessed by the caregiver at any time.

### 3.2.3. *ADL level*

At this level we consider the ADL as evaluated in the Katz scale, in particular:

- transfer (mobility) has the same indicators as balance and gait;
- dressing has the same indicators as balance and visual functionalities;
- feeding has the same indicators as nutrition functionality;
- bathing has no indicators in the current version;
- toileting has no indicators in the current version.

ADLs that are not associated with any indicator cannot be evaluated to identify worsenings. For this reason, only prevention is possible, based on correlation rules (see Section 4 for details).

Mobility is evaluated through the Tinetti-POMA scale while other ADLs are evaluated according to the Katz index. We wish to point out that reasoning at this level is not aimed at activity recognition through the identification of patterns of behaviour, as in other approaches to monitoring [36]. We rather concentrate on possible inter-dependencies that may arise in performing ADLs, according to correlations with items at other levels, because this is useful for prevention.

Instrumental activities of daily living (IADL) from the Lawton scale [37] have not been included. This choice has been guided by the fact that their impact on other health-related items is less determinant and the evaluation with the state-of-the-art sensor technology is too complex to be performed in a nonintrusive way.

*3.2.4. Risk assessment level*
Risks are identified by the potentially most dangerous situations for elderly people at home, namely:

- risk of falls, initially evaluated through the Tinetti-POMA scale; it has the same indicators as balance and gait functionalities;
- risk of depression, initially evaluated through the GDS scale; indicators are those of nutrition, balance, gait and sleep functionalities;
- frailty, initially evaluated through a combination of GDS test, mini mental test and Katz evaluation; indicators are the same as those of nutrition, balance, gait, vision and emotional functionalities plus some additional ones like walking speed, age, number of pathologies, number of drugs and number of activities in which the patient needs help;
- risk of dependency, evaluated through the Katz index; it has the number of ADLs that cannot be easily performed as indicator;
- malnutrition, evaluated through the mini nutritional assessment test with BMI as indicator;
- isolation, having the number of visits and the time spent out of the house as indicators.

## 4. THE REASONING CAPABILITIES

In the knowledge representation model of SINDI we describe a home healthcare scenario by a declarative representation of the domain at different levels.

At the state level, data provided by the sensors network can be noisy and imprecise, even after aggregation. The expressive power of ASP is used to disambiguate unclear situations as much as possible, by using defaults, nondeterministic choice and constraints over the solutions.

Let us consider localization as an example of how the reasoning process helps in the interpretation of context-dependent data.

EXAMPLE 1. SINDI's localization component is based on the intensity variations of the radio signals exchanged between nodes. Unfortunately, it is not always true that the higher the measured intensity of a signal from a node, the closest the person is to that node.

Given proximity values with a certain accuracy $P$ and defined over (possibly overlapping) time intervals $T_i$, $T_j$, the ASP program identifies all possible sequences of moves across rooms and areas. Time intervals are split into temporal segments as follows: whenever two time intervals $T1, T2$ and $T3, T4$ overlap, a splitting point is added in correspondence to the point where overlapping starts and/or ends, dividing the original intervals into sub-intervals we refer to as *segments*. Consider data in Table 3: the localization process returns proximity to the *Bedroom-Bed* area from time 0 to 4 with reliability $P = 10$, proximity to the *LivRoom-Sofa* area from time 3 to 4 with $P = 45$ and proximity to the *Kitchen-Table* area from time 1 to 5 with $P = 45$. The reasoning process generates proximity segments represented by predicates of the form proximity_segm($R$, $A$, $T1$, $T2$, $P$), as follows:

**TABLE 3.** Results of localization for Example 1.

| Time | Bedroom-Bed | LivRoom-Sofa | Kitchen-Table |
|---|---|---|---|
| 0 | 10 | 0 | 0 |
| 1 | 10 | 0 | 45 |
| 2 | 10 | 0 | 45 |
| 3 | 10 | 45 | 45 |
| 4 | 10 | 45 | 45 |
| 5 | 0 | 0 | 45 |

```
proximity_segm(bedroom,bed,0,1,10)
proximity_segm(kitchen,table,1,3,45)
proximity_segm(bedroom,bed,1,3,10)
proximity_segm(bedroom,bed,3,4,10)
proximity_segm(kitchen,table,3,4,45)
proximity_segm(living_room,sofa,3,4,45)
proximity_segm(kitchen,table,4,5,45)
```

Logical rules state that, by default, proximity to an area $A$ of a room $R$ in a temporal segment $T1, T2$ is given by the fact that a signal has been received from the corresponding node in that temporal segment. This holds unless there is a more reliable signal received in the same interval from another node:

```
max_proximity(R,A,T1,T2,P):-
            proximity_segm(R,A,T1,T2,P),
            not other_max(R,A,T1,T2).
```

This other signal determines proximity unless additional contextual data make it invalid (e.g. load cell pressed in a different area $A1$ of room $R1$):

```
other_max(R,A,T1,T2) :-proximity_segm(R,A,T1,
                        T2,P),
                        proximity_segm(R1,A1,
                        T1,T2,P1),
                        P1>P, A!=A1.
                        not invalid(R1,A1,
                        T1,T2).

invalid(R,A,T1,T2) :- proximity_segm(R,A,T1,
                      T2,P),
                      loadcell(R1,A1,Val,T),
                      Val!=0, T1<T<T2, A1!=A.
```

According to proximity values and data available from load cells, results of the computation of consistent proximities for each segment may change. As long as no information about load cells is available, the ASP program computing the maximum consistent proximity in this example will give the following solution:

```
max_proximity(bedroom,bed,0,2,10)
max_proximity(kitchen,table,2,3,45)
max_proximity(kitchen,table,3,4,45)
max_proximity(living_room,sofa,3,4,45)
max_proximity(kitchen,table,4,5,45)
```

Suppose we add predicate

```
loadcell(bedroom,bed,68,2).
```

as a fact indicating that the load cell under the bed has been pressed at time $T = 2$, measuring a weight equal to 68 kg. The new solution is now represented by the following predicates:

```
invalid(kitchen,table,1,3)
max_proximity(bedroom,bed,0,1,10)
max_proximity(bedroom,bed,1,3,10)
max_proximity(kitchen,table,4,5,45)
max_proximity(kitchen,table,3,4,45)
max_proximity(living_room,sofa,3,4,45)
```

When disambiguation is no longer possible and multiple options are available, it can be useful to generate all the alternatives and to reason about them in parallel. This feature turns out to be crucial also for diagnosis and prevention when all possible alternatives are generated according to domain specific knowledge; constraints based on context-dependent data may further reduce the space of the solutions.

EXAMPLE 2. Returning to the situation in Example 1, location based on contextual data and proximity returns two possible results in the interval 3–4. In order to identify the solution that is more consistent with contextual data, it may be necessary to treat both solutions separately. We can do it in ASP by using nondeterministic choice and constraints: a choice rule indicates that for every possible location identified by predicate max_proximity($R$, $A$, $T1$, $T2$, $P$), the person may be there (identified by predicate isIn($R$, $A$, $T1$, $T2$, $P$) or not; the auxiliary predicate aux_isIn($T1$, $T2$) is used to check if there is at least one valid location in the corresponding time interval; integrity constraints in the last two rules specify that a person can be at most in one area for each temporal segment, and each temporal segment must be associated with at least one area when there is a proximity value from the corresponding node:

```
{isIn(R,A,T1,T2,P)}:- max_proximity(R,A,T1,
                      T2,P).

aux_isIn(T1,T2) :- isIn(R,A,T1,T2,P).

:- isIn(R,A,T1,T2,P1), isIn(R1,A1,T1,T2,P2),
A!=A1.

:- not aux_isIn(T1,T2), max_proximity(R,A,T1,
T2,P).
```

The evaluation of the ASP program now returns two solutions with respect to the isIn() predicate:

```
Answer: 1
isIn(bedroom,bed,0,1)
isIn(bedroom,bed,1,3)
isIn(living_room,sofa,3,4)
isIn(kitchen,table,4,5)

Answer: 2
isIn(bedroom,bed,0,1)
isIn(bedroom,bed,1,3)
isIn(kitchen,table,3,4)
isIn(kitchen,table,4,5)
```

A consistent view of the context allows evaluation of indicators that are not directly available from aggregated sensor data. Considering the indicator *quality of sleep* mentioned in the previous section, the ASP code considers a large amount of information to determine a consistent value for this indicator, namely:

- localization details, represented by predicates isIn($R$, $A$, $T1$, $T2$);
- state of the wearable device, represented by predicate w_device($V$, $T$), where value $V$ can be *on* or *off*;
- values returned by load cells, represented by predicate loadcell($R$, $A$, $V$, $T$);

The reasoning process infers additional information referred to:

- beginning of the night period, indicated by predicate nightstart($T1$);
- end of the night period, indicated by predicate nightend($T0$);
- the fact that the person exits bed at time $T$, indicated by predicate exitbed($T$);
- the fact that there is a sleep break at time $T$, indicated by predicate break($T$);
- the fact that the person gets out of the bedroom between time $Ta$ and $Tb$, indicated by predicate out_bedroom($Ta$, $Tb$).

We report part of the logical encoding:

```
nighstart(T1) :- isIn(bedroom,bed,T1,T2),
                 w_device(off,T1).

nightend(T0) :- exitbed(T0), w_device(on,T),
                T0<T,
                not exitbed(T2), T0<T2<T,
                out_bedroom(T3), T<T3.
```

```
exitbed(T) :- loadcell(bedroom,bed,V,T1), V>0,
              loadcell(bedroom,bed,V1,T), V1=0,
              not loadcell(bedroom,bed,V2,T2),
              V=0, T1<T2<T.

break(T) :- nighstart(T1), nightend(T2),
            exitbed(T),T1<T<T2.

out_bedroom(Ta,Tb) :- nighstart(T1),
                      nightend(T2),
                      isIn(R,A,Ta,Tb),
                      T1<Ta<T2.
```

The following logic rules use defaults to discriminate among possible values for the indicator *quality of sleep* (good, medium, bad), according to the information inferred above:

```
obsind(sleep,good,T1,T2) :-
      nightstart(T1), nightend(T2),
      not break(Ta), 1<Ta<T2,
      not out_bedroom(Tb,Tc), T1<Tb<T2.

obsind(sleep,medium,T1,T2) :-
      nightstart(T1), nightend(T2),
      break(T), T1<Ta<T2,
      not out_bedroom(Tb,Tc), T1<Tb<T2.

obsind(sleep,bad,T1,T2) :-
      nightstart(T1), nightend(T2),
      break(T), out_bedroom(Tb,Tc),
      T1<Ta<T2, T1<Tb<T2.
```

In order to deal with emergencies, SINDI can be configured to detect some triggers. Such triggers can either generate a direct action (e.g. an emergency call) or rely on the reasoning system. As an example, temperature over 40°C is set as a trigger for an emergency call, while the opening of a window needs to be contextualized using specific rules in order to check whether it is an intrusion or not.

At upper levels, inference is performed by separate logic programs in order to detect:

(i) functional disabilities, every hour;
(ii) dependencies in performing ADL, every day;
(iii) risk assessment, every day.

Besides domain knowledge and consistent interpretation of the context, two more aspects are necessary in order to reason about the health status of the person monitored: *differential evaluations* and *correlation rules*.

Differential evaluation of an item $I$ at level $L$ through the indicators $\mathrm{Ind}_i$ is possible by comparing the value $V_i^0$ of each associated indicator at the beginning of the previous inference (time 0) and the (eventually aggregated) value $V_i^1$ of the same indicator at the time interval being evaluated (time 1).

For some indicators such as standing and sitting, several evaluations may be available for the time interval (hour or day) considered in the inference process. Given that a single value has to be provided for each indicator in a given interval, the data extraction module taking data from the database and passing them to the ASP engine is in charge of computing the most frequent value for that interval. This choice can be motivated by the fact that the slow trend of physical and cognitive decline makes evaluations uniform in a short period of time such as an hour or a day, and isolated values that are far from the most frequent one can be due to occasional awkward movements rather than to a disability.

Though differential evaluations can also indicate improvements, we only consider worsenings, as they are much more relevant with respect to risk prevention. The reasoning system can be extended to also consider health improvements and use them to evaluate response to medical treatments.

Worsenings can be detected applying the following logic rule:

```
worse(L,I,Ind,T1,T2) :- obsind(Ind,V,T1-1,T1),
                        obsind(Ind,V1,T1,T2),
                        link(I,Ind), lev(L,I),
                        ord(Ind,V1,N1), T1<T2,
                        ord(Ind,V,N), N<N1.
```

Correlation rules concern dependencies between a cause ($I_1$) and an effect ($I$). Different dependencies are allowed:

- only negative influence of an item $I_j$ on another item $I_k$;
- only positive influence of an item $I_j$ on another item $I_k$;
- directly proportional influence of an item $I_j$ on another item $I_k$;
- inversely proportional influence of an item $I_j$ on another item $I_k$;

Each of these correlations can be *strict* or *possible*. In the first release of the system we concentrated on strict and possible negative influence, since they are more significant for prevention and diagnosis. All other dependencies can be introduced and encoded in the system in a similar way, and we are considering this issue in the implementation of the following prototype of SINDI.

Correlation rules can be specified by clinicians and automatically mapped into ASP to express negative/possibly negative influence of an item $I_k$ on another item $I_j$, respectively encoded into logic predicates:

```
neg_influence(I_j,I_k).
poss_neg_influence(I_j,I_k).
```

Consider the structure of items and correlation rules as an oriented graph stratified into levels: items are nodes and correlations are oriented arcs connecting two nodes.

Each item $I_j$ at a levels $L$, $I_j(L)$, can be connected to another item $I_k(L')$ in two different ways:

- oriented arc from $I_j(L)$ to $I_k(L')$: if $I_j(L)$ gets worse, this has negative influence on $I_k(L')$;
- oriented dotted arc from $I_j(L)$ to $I_k(L')$: if $I_j(L)$ gets worse, this *may* have negative influence on $I_k(L')$.

In addition, the layered structure is used to avoid possibly infinite propagation of dependencies when the reasoning process investigates the search space.

As already mentioned, the inference process considers items at each level separately. No matter which level is being evaluated, the system first characterizes every item in the graph as being either *stable* (none of the indicators become worse for that item) or *unstable* (one or more indicators become worse for that item in the interval being evaluated). When at least one indicator becomes worse, the item is marked as unstable, otherwise it is stable:

```
unstable(L,I,T1,T2) :- worse(L,I,Ind,T1,T2),
                       lev(L,I).

stable(L,I,T1,T2) :- not unstable(L,I,T1,T2),
                     lev(L,I).
```

This distinction is crucial to determine the behaviour of the system when it reasons about each $I_j(L)$ at the specific level $L$:

(a) if $I_j(L)$ is *stable*, the system performs the following reasoning task:
   - makes predictions about the amount of risk for $I_j(L)$ to get unstable, as follows:
     - investigates the direct connections determined by correlations rules, to identify items $I_k(L')$ that may influence $I_j(L)$;
     - check each $I_k(L')$ to see whether it is unstable and, in this case, conclude that $I_j(L)$ can be at risk due to its correlation to $I_k(L')$;

(b) if $I_j(L)$ is *unstable*, the system performs three different reasoning tasks at once:
   - identifies possible negative effects of the worsening of $I_j(L)$ on other items $I_k(L')$ according to correlation rules, represented by oriented arcs from $I_j(L)$ to $I_k(L')$[1];
   - performs *local* diagnosis, detecting possible alternative causes of the worsening of $I_j(L)$ among the correlated items that have been marked as unstable;
   - contextualizes the worsening of $I_j(L)$ providing alternative health-related contextualizations and values of items (both stable and unstable ones) included in each separate contextualization scenarios.

[1]Propagation of negative effects are not considered since the layered structure of the graph lets us identify them simply by investigating results of the inference for $I_k(L')$ when items at level $L'$ are evaluated.

Note that diagnosis and contextualization reasoning tasks may generate alternative solutions. This is due to the fact that two items $I_1$, $I_2$ that are related to the item being investigated $I$, can be in the same path backwards from $I$ or not. In the second case, they are part of two different solutions.

Just as an example, we report part of the encoding used for prediction of functional disabilities, with respect to a stable function $F$. This corresponds to point (a) in the description of the algorithm. Logic rules make it possible to predict negative effects of the worsenings of $I$ on function $F$:

```
% Prediction
poss_neg_pred(func,F,L,I,T1,T2) :-
                stable(func,F,T1,T2),
                poss_neg_influence(F,I),
                unstable(L,I,T1,T2).
neg_pred(func,F,L,I,T1,T2) :-
           stable(func,F,T1,T2),
           neg_influence(F,I),
           unstable(L,I,T1,T2).
```

To conclude this section, we present a simple example on how the reasoning tasks can support clinicians in health assessment of the elderly.

EXAMPLE 3. Consider the graph in Fig. 9. Suppose that the reasoning system is investigating ADL dependencies (level 2) and items are stable at this level (light grey nodes). Suppose also that *visual* functionality is marked as unstable (black node). Results of the inference process with respect to prediction are illustrated in Fig. 10: ADLs' *dress* and *eat* are both at risk (dark grey nodes) due to the *visual* functionality, but in one case (for the ADL *eat*) the risk is only *possible*. This qualitative interpretation allows the association of worsenings to priorities and guides caregivers in planning interventions.

EXAMPLE 4. Suppose now that, in the same setting, the initial evaluation of ADL dependencies is the one depicted in

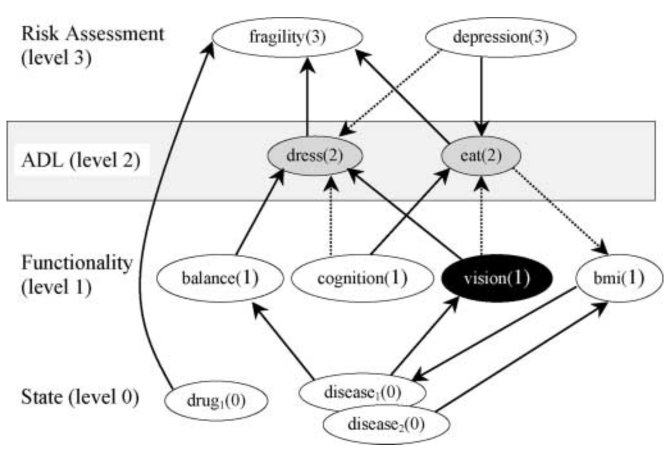


**FIGURE 9.** Example 3: correlations and graph colouring.

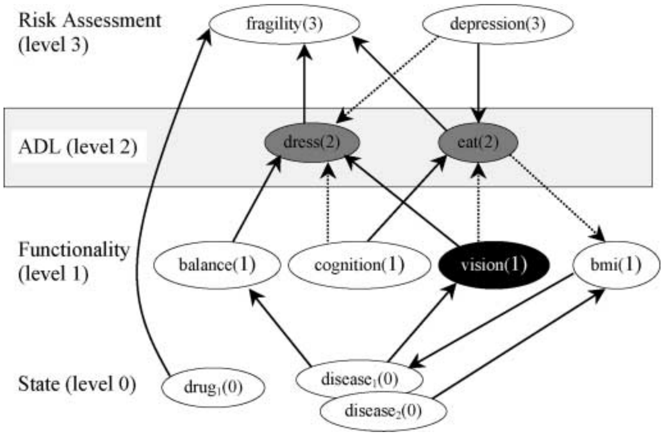


**FIGURE 10.** Example 3: results of prediction task.

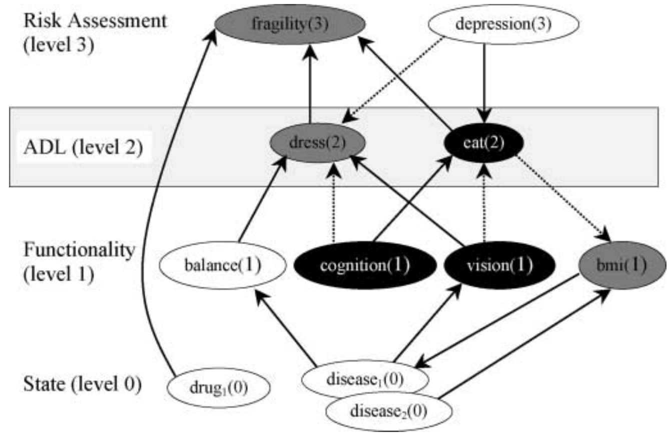


**FIGURE 12.** Example 4: results of prediction task.

Fig. 11: ADL *eat* is unstable (i.e. there is an increasing level of dependency in performing *eat*), and *visual* and *cognitive* functionalities are unstable too. The inference process returns the following results:

- *Prediction*: risk of fragility and nutrition functionality (represented by the BMI) have to be monitored carefully since they may get worse due to dependency in performing *eat* (Fig. 12); the system also returns that ADL *dress* is at risk due to cognitive and visual functionalities;
- *Local diagnosis*: the increased level of dependency in performing *eat* can be due to a functional disability in vision and/or cognition (Fig. 13); these possible explanations are treated as separate solutions since following a path from *eat* backward, leads to either one or the other functionality;
- *Contextualization*: following the links backwards from the unstable ADL *eat*, three contextualization sets are identified for its worsening, as shown in Fig. 14; values of items and related indicators are provided to clinicians through appropriate interfaces, helping to identify the most plausible alternative contextualizations of the worsening of ADL *eat* (Fig. 14) according to the context.

It is easy to figure out how, in a more complex schema of dependencies among items, loops and multiple paths can make reasoning a hard task. For this reason we decided to make the graph structure hierarchical, in that reasoning tasks for items at one level are performed separately from the reasoning tasks for items at other levels. This distinction is both conceptual and temporal, since inference is run every hour for functional evaluation and every day for ADL dependency and risk assessment. In dealing with complex graphs of dependencies, the expressive power of ASP can be crucial since it makes it possible to explore complex search spaces maintaining the computational complexity rather low.

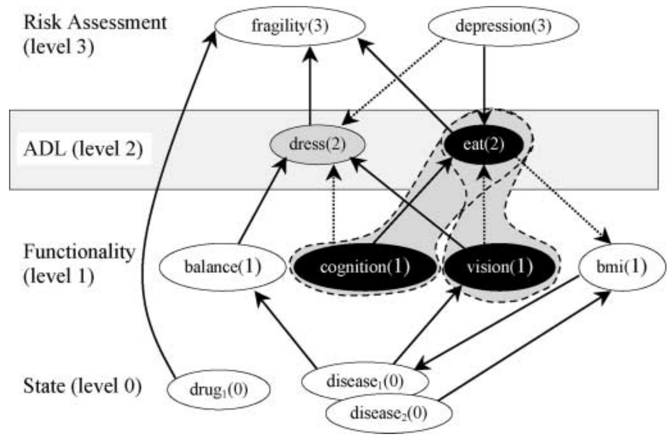


**FIGURE 13.** Example 4: results of diagnosis task.

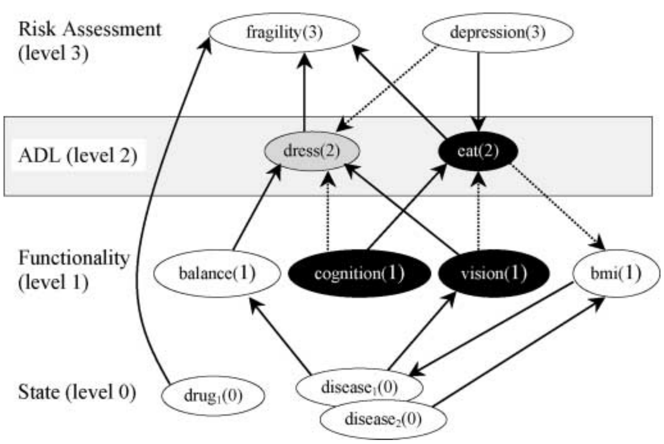


**FIGURE 11.** Example 4: correlations and graph colouring.

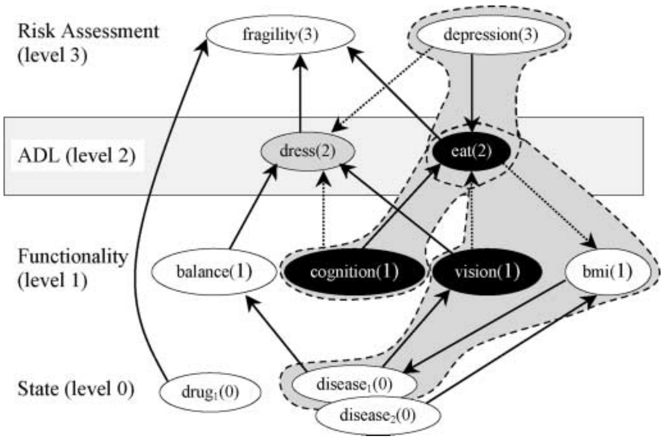


**FIGURE 14.** Example 4: results of contextualization task.

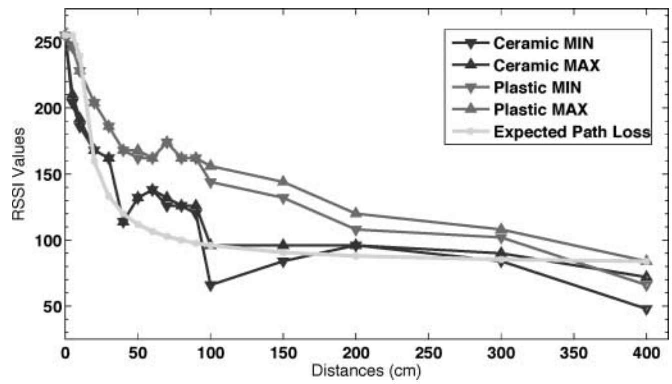


**FIGURE 15.** Test 1: RSSI values and distance.

## 5. PRELIMINARY EVALUATION

The evaluation of an assisted-living system like SINDI is very difficult because it is hard to identify the correct metrics. The need for a common framework to identify the challenges and to suggest the metrics is clear. The work described in [38] proposes an *evaluation framework* for assessing the quality of assistive environments. This framework identifies a set of attributes that are considered critical for user adoption. The categories identified are the following:

- functionality (correctness, robustness—errors and faults, reliability—time of continuous operation);
- usability (ease of use, nonobtrusiveness, accessibility);
- security and privacy (such as access modes and encryption);
- architecture (modularity, interoperability—standard interfaces to integrate components);
- cost (installation, maintenance);

This framework could be improved by using separate evaluation metrics for users and technical experts.

So far we have run the full system for short periods of time (days) in a mock-up environment without real users. Nonobtrusiveness stems from the design of SINDI. Details about the user and the environment are automatically collected by the WSN, and no complex statistical information or specific medical knowledge is needed to analyse possible evolutions of patient's health and to support understanding. Moreover, the interaction of SINDI with the patient and the caregivers is fully intuitive, as we deal with multimedia contents and the patient is provided with a device that looks like a remote control. Security and privacy are guaranteed by the use of security standards and techniques. Modularity and computational efficiency stem from the declarative nature of ASP and the availability of efficient solvers. Finally, the use of off-the-shelf components in SINDI considerably reduces overall costs.

In the preliminary evaluation we did several tests on the WSN and on the inference engine. Details are presented in the following subsections.

### 5.1. WSN evaluation

#### 5.1.1. *Localization and tracking system evaluation*

A preliminary evaluation of the localization and tracking system has been done configuring a test environment composed by a master processor, a wearable device, two base nodes and six environmental nodes assigned to two logical zones.

In the first test, we recorded the RSSI values of many messages between two nodes of the network at various distances (10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 300, 400 cm). The nodes have two different types of antenna (ceramic and plastic) and different battery charge levels (high–medium–low).

The test results are shown in Fig. 15. This test shows that: (i) the RSSI values are closely tied to environmental conditions (such as location of the sensors in the room, presence of people, motion sensors), (ii) the RSSI values are independent of battery charge-levels, (iii) the type of antenna considerably affects the RSSI values. For a general assessment of the behaviour of RSSI see [39].

The aim of the second test is to evaluate the boundary accuracy of the localization system during a zone change. We used the same network setting described above and we marked an imaginary boundary between the two zones. We recorded many RSSI values at various distances from the boundary (−200, −150, −50, 0, 50, 100, 150 cm) by asking a person to walk back and forth on an approximately straight line between the two zones. We repeated the test 30 times.

The results are plotted in Fig. 16. The localization algorithm can recognize the correct zone 90% of the time without further filtering techniques.

#### 5.1.2. *Movement recognition evaluation*

Tests on movement recognition were done with 20 people aged between 21 and 55. We worked under realistic conditions, and

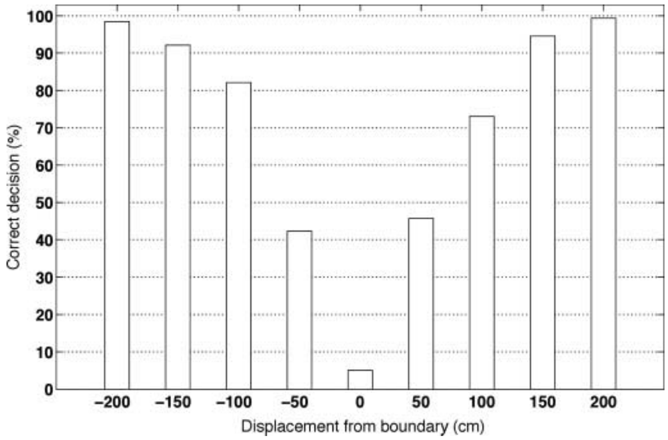


**FIGURE 16.** Test 2: evaluation of the boundary accuracy of the localization system.

**TABLE 4.** Confusion matrix.

| | a | b | c | d | e | f | g | h | i | z |
|---|---|---|---|---|---|---|---|---|---|---|
| a | 102 | – | – | – | – | – | – | – | – | 7 |
| b | 6 | 83 | – | – | – | – | – | – | – | – |
| c | – | – | 40 | 1 | – | – | – | – | – | – |
| d | – | – | 1 | 38 | 3 | – | – | – | – | – |
| e | – | – | – | 2 | 41 | – | – | – | – | – |
| f | – | – | – | – | – | 82 | – | – | – | 2 |
| g | – | – | – | – | – | – | 53 | – | – | 1 |
| h | – | – | – | – | – | – | – | 74 | – | 3 |
| i | – | – | – | – | – | – | – | – | 16 | 4 |

(a) Walking, (b) staying, (c) lying, (d) sitting, (e) standing, (f) turning right, (g) turning left, (h) turning back, (i) fall on a sit (z) others. Column labels represents the ground truth.

we did our activity recognition at runtime. We used just a single source of information—a single triaxial inertial sensor—worn on the hip of the person. The algorithms used in SINDI do not need to be trained. We show our test results of the activity recognition algorithms with a confusion matrix (see Table 4).

The activities we wish to recognize are walking, staying, sitting, lying, standing, turning right, left, back and falling. Every person was asked to do a specific set of activities in a natural way to have at least two instances of every activity for each person. Because of the difficulty of simulating falls with real people we asked them to do a backward fall on a chair; this test is not as accurate as we would like because it is not a 'real fall', but we put it in the confusion matrix table for completeness; we plan to repeat this experiment using a test dummy. We achieved accuracy rates of 88–96% for walking, and staying activities, of 95–98% for standing still, sitting, lying activities, 97–98% for turnings and an accuracy of 67–70% for fall detections (fall on a chair), with an overall performance of 88.5% . The test must be extended, especially for the fall detection case, to a greater number of people, but considering the context of use these results seem reasonable for movements recognition in realistic conditions with inertial sensors.

**TABLE 5.** ASP reasoning performance (time is expressed in seconds).

| Items (No.) | State level (time) WSN/ASP vs. ASP | Correlations (No.) | Upper levels (time) |
|---|---|---|---|
| 20 | 140.05 vs. 205.07 | 30 | 0.69 |
| | | 70 | 1.01 |
| | | 130 | 2.03 |
| 30 | 169.71 vs. 481.22 | 120 | 1.03 |
| 40 | 183.48 vs. 487.17 | 160 | 1.11 |
| 50 | 201.32 vs. 589.03 | 200 | 1.37 |
| 70 | 241.11 vs. 603.16 | 250 | 1.68 |

### 5.2. Inference engine evaluation

In the first testing environment of our system we evaluated ASP programs using *Lparse* as grounder and the *Clasp* solver [24, 40] as inference engine.

The Clasp solver supports constraints, choice rules and weight rules [41] and it can solve complex reasoning tasks very efficiently due to the heuristics used, combining ASP expressivity with boolean constraint solving.

In the testing phase of SINDI, we used Clasp both to generate the backlog (a few months of data) and to test the global performance of the system.

We did some tests on randomly generated instances with 20–70 correlation rules and 10–50 items, obtaining results in 0.65–2.75 s, once state level data had been aggregated and interpreted correctly. The worst cases were observed for instances where the number of correlations was more than six times the number of items. These results are due to the high number of bidirectional correlations among items, derived by the random generation of instances. According to geriatric practitioners, similar cases are not common in real settings and, except for those instances, the reasoning process scales well. Times of execution in sample instances are summarized in Table 5.

Context aggregation and interpretation at the state level remains the harder task, since it requires to analysis of up to 24 h of data when reasoning about ADL dependency and risk assessment is performed. In evaluating indicators, delegating part of the aggregation process to the WSN nodes lowered the computational time up to 60% for instances of medium complexity (i.e. for a person who is active from 30% to 40% of the time in a day).

These computational costs do not apply to situations in which emergencies arise, since they are detected almost immediately by triggering events and managed by evaluating appropriate integrity constraints.

## 6. CONCLUSIONS AND FUTURE WORK

The solution we propose for the delivery of clinical care is based on state-of-the-art WSN technology that allows cheap and constant monitoring of a patient, together with efficient reasoning techniques aimed at preventing risky situations before they arise.

The logical ASP approach is unobtrusive, modular, declarative [42] and efficient. Nonintrusiveness is granted by the fact that information about the user is extracted automatically by the WSN of SINDI: no complex statistical information or general medical knowledge is needed to determine the nature of the emergency. Modularity is given by the default reasoning and a declarative specification of the problem, while efficiency relies on the quality of the ASP implementation.

The interaction of the whole system with the patient and the caregivers is fully intuitive, since it is based on multimedia contents and the patient is provided with a handheld device that works like a TV remote.

SINDI's reasoning process encodes both medical and commonsense knowledge. While commonsense rules are specified by a knowledge engineer, medical knowledge stems from the formalization of specific medical scales. We believe that the declarative approach we propose could make it possible for clinicians to specify their own scales on the basis of the available sensors. In this respect, we are investigating the specification of a high-level language that makes it possible to specify additional medical scales in action description language fashion, so that these specifications can be then automatically mapped into a logic ASP program and used in the inference process.

A further issue is related to the outputs (in terms of evaluation of risky situations) provided by the system. One of the interesting aspects of using ASP semantics in this context is that all possible correlations among factors of different levels are considered equally important and valid. There are efficient techniques to enforce priorities and ordering relations among solutions of an ASP program [43, 44], and it would be interesting to investigate how to apply these techniques in healthcare applications.

The graphical representation of dependencies and results of the reasoning tasks at any timestamp suggest that automatic methods can be applied to the analysis of the history of inferences. We wish to investigate these issues to include them in the following release of the system.

Preliminary tests showed that the system could be profitably employed by home healthcare services supporting the delivery of care. Nonetheless, we are aware of the fact that more detailed and extensive experimental results are needed to evaluate the effectiveness of this approach in different social contexts, and to provide significant empirical data.

## FUNDING

This research has been funded by the Qua_SI project (www.quasi.unimib.it) and the Italian Ministry of Research.

## REFERENCES

[1] Giuliani, M.V., Scopelliti, M. and Fornara, F. (2005) Elderly People at Home: Technological Help in Everyday Activities. *Proc. ROMAN 2005*, Nashville, TN, August 13–15, pp. 365–370. IEEE Computer Society, Washington, DC, USA.

[2] Dishman, E. (2004) Inventing wellness systems for aging in place. *IEEE Comput.,* **37**, 34–41.

[3] Teich, J., Schmiz, J., Kuperman, G., and Bates, D. (2000) Effects of computerized physician order entry on prescribing practices. *Arch. Intern. Med.*, **160**, 2741–2747.

[4] Akyildiz, I.F., Weilian, S., Sankarasubramaniam, Y. and Cayirci, E. E. (2002) A survey on sensor networks. *IEEE Commun. Mag.*, **40**, 102–114.

[5] Li, M. and Yang, B. (2006) A Survey on Topology Issues in Wireless Sensor Network. *Proc. ICWN 2006*, Las Vegas, NV, USA, June 26–29, pp. 503–509. CSREA Press, Washington, DC, USA.

[6] CS-2006-11 (2006) *ALARM-NET: wireless sensor networks for assisted-living and residential monitoring*. Department of Computer Science, University of Virginia, USA.

[7] Al-Omari, S. and Shi, W. (2007) Toward Highly Available WSNs for Assisted Living. *Proc. ACM/SIGMOBILE HealthNet 2007*, San Juan, Puerto Rico, June 11–14, pp. 31–36. ACM Press, New York, NY, USA.

[8] Malan, D., Fulford-Jones, T., Wesh, M. and Moulton, S. (2004) CodeBlue: An Ad hoc Sensor Network Infrastructure for Emergency Medical Care. *Proc. WAMES 2004*, MA, USA, June 6, pp. 12–14. ACM Press, New York, NY, USA.

[9] Kochhal, M., Schwiebert, L. and Gupta, S. (2003) Role-based Hierarchical Self Organization for Wireless Ad hoc Sensor Networks. *Proc. WSNA 2003*, San Diego, CA, USA, September 19, pp. 98–107. ACM Press, New York, NY, USA.

[10] Heinzelman, W., Chandrakasan, A. and Balakrishnan, H. (2000) Energy-efficient Communication Protocol for Wireless Sensor Networks. *Proc. HICSS 2000*, Maui, HI, USA, January 4–7, pp. 8020–8029. IEEE Computer Society, Washington, DC, USA.

[11] Manjeshwar, A. and Agrawal, D.P. (2001) TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. *Proc. IPDPS 2001*, San Francisco, CA, April 23–27, pp. 2009–2015. IEEE Computer Society, Washington, DC, USA.

[12] Manjeshwar, A. and Agrawal, D. P. (2002) APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. *Proc. IPDPS 2002*, Fort Lauderdale, FL, April 15–19, pp. 195–202. IEEE Computer Society, Los Alamitos, CA, USA.

[13] Souto, E., Guimarães, G., Vasconcelos, G., Vieira, M., Rosa, N. and Ferraz, C. (2004) A Message-oriented Middleware for Sensor Networks. *Proc. MPAC 2004*, Toronto, ON, Canada, October 18–22, pp. 127–134. ACM, New York, NY, USA.

[14] Costa, P., Picco, G. and Rossetto, S. (2005) Publish-subscribe on Sensor Networks: A Semi-probabilistic Approach. *Proc. MASS*

*2005*, Washington, DC, November 7–9, pp. 332–342. IEEE Computer Society, Los Alamitos, CA, USA.

[15] Krishnamachari, B., Estrin, D. and Wicker, S. B. (2002) The Impact of Data Aggregation in Wireless Sensor Networks. *Proc. ICDCS 2002*, Vienna, Austria, July 2–5. IEEE Computer Society, Washington, DC, USA.

[16] Lester, J., Choudhury, T., Borriello, G., Consolvo, S., Landay, J., Everitt, K. and Smith, I. (2005) Sensing and Modeling Activities to Support Physical Fitness. *Proc. Ubicomp 2005*, Tokyo, Japan, September 10–11, pp. 1–4. Springer, Berlin.

[17] Bao, L. and Intille, S.S. (2004) Activity Recognition From User-annotated Acceleration Data. *Proc. Pervasive 2004*, Vienna, Austria, April 21–23, pp. 1–17. Springer, Berlin.

[18] Bao, L. (2003) Physical activity recognition from acceleration data under semi-naturalistic conditions. Master's Thesis, Department of Electrical Engineering and Computer Science, MIT Institute, MA, USA.

[19] Lester, J., Choudhury, T. and Borriello, G. (2006) A Practical Approach to Recognizing Physical Activities. *Proc. Pervasive 2006*, Dublin, Ireland, May 7–10, pp. 1–16. Springer, Berlin.

[20] Mantyjarvi, J., Himberg, J. and Seppanen, T. (2001) Recognizing Human Motion with Multiple Acceleration Sensors. *Proc. SMC 2001*, Tucson, AZ, USA, October 7–10, pp. 747–752. IEEE Computer Society, Washington, DC, USA.

[21] Gelfond, M. and Lifschitz, V. (1988) The Stable Model Semantics for Logic Programming. *Proc. ICLP 1988*, Seattle, Washington, August 15–18, pp. 1070–1080. MIT Press, MA, USA.

[22] Simons, P., Niemelä, I. and Soininen, T. (2002) Extending and implementing the stable model semantics. *Artif. Intell. J.*, **138**, 181–234.

[23] Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S. and Scarcello, F. (2006) The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.*, **7**, 499–562.

[24] Gebser, M., Kaufmann, B., Neumann, A. and Schaub, T. (2007) Conflict-driven Answer Set Solving. *Proc. IJCAI 2007*, Hyderabad, India, January 6–12, pp. 386–392. AAAI Press, MA, USA.

[25] Web pages of the most efficient solvers for ASP. Clasp: http://sourceforge.net/projects/potassco/, Smodels: http://www.tcs.hut.fi/Software/smodels/, PSmodels: http://www.tcs.hut.fi/Software/smodels/priority/, DLV: http://www.dbai.tuwien.ac.at/proj/dlv/, Cmodels: http://www.cs.utexas.edu/∼tag/cmodels/.

[26] HUT-TCS-Y2001 (2001) *Answer-set programming: a declarative knowledge representation paradigm*. Helsinki University of Technology, Helsinkj, Finland.

[27] Baral, C. and Gelfond, M. (2002) Logic programming and knowledge representation—the A-prolog perspective. *Artif. Intell. J.*, **138**, 3–38.

[28] Fleming, Kevin C., Evans, J.M., Weber, D.C. and Chutka, D.S. (1995) Practical functional assessment of elderly persons: a primary-care approach. *Mayo Clin.*, **70**, 890–910.

[29] Folstein, M.F., Folstein, S.E. and McHugh, P.R. (1975) Mini-mental state: a practical method for grading the cognitive state of patients for the clinician. *J. Psychiatr. Res.*, **12**, 189–198.

[30] Yamamoto, S., Mogi, N., Umegaki, H., Suzuki, Y., Ando, F., Shimokata, H. and Iguchi, A. (2004) The clock drawing test as a valid screening method for mild cognitive impairment. *Dement. Geriatr. Cogn. Disord.*, **18**, 172–179.

[31] Tinetti, M., Williams, T. and Mayewski, P. (2002) Fall risk index for elderly patients based on number of chronic disabilities. *Amer. J. Med.*, **80**, 429–434.

[32] Yesavage, J., Brink, T., Rose, T., Lum, O., Huang, V., Adey, M. and Leirer, V. (1982–1983) Development and validation of a geriatric depression screening scale: a preliminary report. *J. Psychiatr. Res.*, **17**, 37–49.

[33] Guigoz, Y., Vellas, B. and Garry, P. (1997) Mini nutritional assessment: a practical assessment tool for grading the nutritional state of elderly patients. *Facts Res. Gerontol.*, **2**, 15–59.

[34] Katz, S., Downs, T., Cash, H. and Grotz, R. (1970) Progress in the development of ADL. *Gerontol.*, **10**, 20–30.

[35] Jitapunkul, S., Pillay, I. and Ebrahim, S. (1991) The abbreviated mental test: its use and validity. *Age Ageing*, **20**, 332–336.

[36] Pollack, M.E. (2005) Intelligent technology for an aging population: the use of AI to assist elders with cognitive impairment. *AI Mag.*, **26**, 9–24.

[37] Lawton, M. (1988) Scales to measure competence in everyday activities. *Psychopharmacol. Bull.*, **24**, 609–614.

[38] Metsis, V., Le, Z., Lei, Y. and Makedon, F. (2008) Towards an Evaluation Framework for Assistive Environments. *Proc. PETRA 2008*, Athens, Greece, July 16–18, pp. 1–8. ACM Press, New York, NY, USA.

[39] Ramadurai, V. and Sichitiu, M.L. (2003) Localization in Wireless Sensor Networks: A Probabilistic Approach. *Proc. ICWN 2003*, Las Vegas, NV, USA, June 23–26, pp. 275–281. CSREA Press, Washington, DC, USA.

[40] Gebser, M., Kaufmann, B., Neumann, A. and Schaub, T. (2007) Clasp: A Conflict-driven Answer Set Solver. *Proc. LPNMR 2007*, Tempe, AZ, USA, May 15–17, pp. 260–265. Springer, Berlin.

[41] Niemelä, I. and Simons, P. (2000) Extending the Smodels System with Cardinality and Weight Constraints. In Minker, J. (ed.), *Logic-based Artificial Intelligence*, pp. 491–521. Kluwer Academic Publishers, Norwell, MA, USA.

[42] Baral, C. (2003) *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, New York, NY, USA.

[43] Brewka, G. (2002) Logic Programming with Ordered Disjunction. *Proc. AAAI 2002*, Edmonton, Alberta, Canada, July 28–August 1, pp. 100–105. AAAI Press, Menlo Park, CA, USA.

[44] Brewka, G., Niemelä, I. and Syrjänen, T. Implementing Ordered Disjunction Using Answer Set Solvers for Normal Programs. *Proc. JELIA 2002*, Cosenza, Italy, September 23–26, pp. 444–455. Springer, Berlin.