International Journal on Artificial Intelligence Tools
Vol. XX, No. X (2013) 1–20
© World Scientific Publishing Company

A MULTI-AGENT CARE SYSTEM TO SUPPORT INDEPENDENT LIVING

BOŠTJAN KALUŽA, BOŽIDARA CVETKOVIĆ, ERIK DOVGAN, HRISTIJAN GJORESKI, MATJAŽ GAMS, MITJA LUŠTREK

Department of Intelligent Systems, Jožef Stefan Institute,
Jamova cesta 39, 1000 Ljubljana, Slovenia
{bostjan.kaluza, boza.cvetkovic, erik.dovgan, hristijan.gjoreski, mitja.lustrek, matjaz.gams}@ijs.si

VIOLETA MIRCHEVSKA

Result, d.o.o., Celovška cesta 182, 1000 Ljubljana,
Slovenia
violeta.mircevska@ijs.si

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

This paper presents a context-aware, multi-agent system called “Confidence” that helps elderly people remain independent longer by detecting falls and unusual movement, which may indicate a health problem. The system combines state-of-the-art sensor technologies and four groups of agents providing a reliable, robust, flexible monitoring system. It can call for help in case of an emergency, and issue warnings if unusual behavior is detected. The first group gathers data from the location and inertial sensors and suppresses noise. The second group reconstructs the position and activity of a person and detects the context. The third group assesses the person’s condition in the environment and reacts to critical situations such as falls. The fourth group detects unusual behavior as an indicator of a potential health problem. The system was successfully tested on a scenario consisting of events that were difficult to recognize as falls, as well as in a scenario consisting of normal days and days when the person was ill. It was also demonstrated live several times, with excellent performance in complex situations.

Keywords: multi-agent system; context awareness; elderly health care; fall detection; detection of health deterioration; ambient assisted living.

1. Introduction

The populations in developed societies are aging rapidly, which threatens to overwhelm society’s capacity for taking care of the elderly. The percentage of people aged 65 and above in the European Union is projected to rise from 17.4 percent in 2010 to 28.8 percent in 2050.[1] As a result, there will be fewer than two people of working age (20–64) for every person above age 65. Such projections drive the urgent need for development of ambient assisted-living solutions to help the elderly live independently for a longer time, with minimal support of the working-age population. Given the choice, most elderly people would prefer to continue to live in their own homes (aging in place).[2] Unfortunately, many elderly people gradually move to an eldercare facility before they completely lose the ability to live independently. The main reason for this is that they fear suffering an accident, such as a fall, and not being able to help themselves. This decreases their quality of life, even if they do not have an accident. Moreover, elderly people living alone can develop health problems that go unnoticed until they become serious. This phenomenon has serious social and economic consequences.

Several systems were recently introduced to address some of the issues related to eldercare. However, most systems developed in research projects are either too expensive for mass use or of too low a quality for real-life circumstances. The commercial solutions are usually based on fall detection. This means that they are capable of recognizing simple hazardous situations.

This paper aims to augment the scope of hazardous-situation detection by enriching it with more complex schemes for reconstructing and interpreting the user’s situation. We are particularly introducing intelligent agents for eldercare, combined with inertial and location sensors. We are studying how these sensors can be employed in a context-aware system capable of robust, reliable monitoring of an elderly person, and detecting simple and complex falls, and unusual gait or movement between rooms. Not only will the elderly benefit from the system, but so will their families and caregivers. Family, caregiver, and societal burden will be substantially reduced. The system will decrease the need for institutionalizing the elderly, and thus reduce healthcare expenditure. This is the central intention of the EU FP7 project Confidence, of which our system is part.[3]

In this context, we are focusing on integrating state-of-the-art technology in the fields of multi-agent systems, context-awareness, filtering, machine learning, and outlier detection to produce a multi-functional system for home eldercare. The system composes a multitude of agents, each of which provides a set of services. The agents are arranged hierarchically, providing increasingly more abstract situational awareness and, in parallel, exploiting the principle of multiple knowledge by asynchronously obtaining heterogeneous information and opinions from several agents.[4]

This paper looks at four groups of agents in a complex system consisting of several more layers. The *refining agents* receive location and inertial measurement messages, filter the noise and map the raw data to the human-body model. The *reconstruction agents* reconstruct the person's 3-D position and activity. The *interpretation agents* provide the system with awareness of dangerous situations by explaining the reconstruction results and reacting to hazardous situations. Finally, the *prevention agents* monitor a variety of measurable parameters, ranging from activity characteristics to daily activities, and react to unusual behavior, health deterioration, or a disease.

Section 2 of this paper reviews related work that addresses the issue of eldercare. Section 3 describes the requirements for the integrated system and the basic implementation choices of the infrastructure, and provides an overview of the agent architecture. Section 4 explains the agents groups in detail, while Section 5 shows an example of how all the agents work together in real life. Section 6 discusses deployment aspects and limitations of the study, and, finally, Section 7 concludes the paper.

2. Eldercare and Fall Detection

Fall detection is a major challenge in eldercare systems, since falls are usually the main cause for hospitalization. Older adults are five times more likely to be hospitalized for a fall-related injury than for any other cause.[5] Fall injuries account for 40 percent of all nursing-home admissions.[6] Pan et al. presented a homecare service that uses a tri-axial accelerometer worn on the body that reports falls to an emergency center.[7] The paper introduces a fall-detection classifier, based on a neural network and a multi-agent architecture for requesting emergency services.

Bourke and Lynos presented a method for detecting falls, using a threshold algorithm and a bi-axial gyroscope sensor mounted on the body that measures pitch and roll angular velocities.[8] Doughty et al. and Kangas et al. reported on fall-detection systems based on a tri-axial accelerometer using threshold value for raising an alarm.[9,10] Willis applied dynamic belief networks to detect falls.[11] Zangh et al. experimented with a support-vector machine classifier.[12] There were also efforts to detect falls by video signals. For example Fu et al. used a temporal-contrast video sensor.[13] Fall-detection systems are also already commercially available. The iLife™ employs a panic button, accelerometers to distinguish between falls, normal movement, and accidental drops of the sensor, and falls reported with wireless transmitter.[14] Zenio provides a similar product equipped with Bluetooth, so it can be connected to a PC or cell phone.[15]

Reports of 100-percent accurate fall detection may lead one to believe that this problem has been solved. However, the reason for such high accuracies is that the methods are usually tested on typical falls in laboratories, which are easy to recognize. By contrast, Li et al. tested for quickly laying down and two atypical falls.[16] They used two tri-axial accelerometers and gyroscopes worn on the chest and thigh. They detected potential falls and activities after the fall by applying thresholds to accelerations, angular velocities, and angles. This resulted in a 90.1 percent fall-detection accuracy rate.

Agent-based systems have recently gained great importance in the healthcare field. Isern et al. reviewed several agent-based solutions. They concluded that such an approach has positive effects in terms of modularity, efficiency, decentralization, flexibility, personalization, distributed planning, monitoring, pro-activity, and security.[17] In the remote care field, they gave special mention to the Agineru system.[18] This is a multi-agent system that monitors and processes physiological parameters on a portable device and enables physicians to access the data through a web application based on a centralized database. Koutkias et al. presented a multi-agent system for managing chronic diseases that detects anomalous cases and informs personnel.[19] Cervantes et al. proposed a similar system for pervasive management, which collects and evaluates physiological data to detect a person's symptoms and report the situation to the doctor.[20] In addition, Moreno et al. reported an agent-based system for proactive monitoring of palliative patients.[21,22] The system monitored patient's health status from self-evaluation reports and doctor's evaluations, detected potential problems, and automatically notified medical personnel.

Camarinha-Matos et al. reported on the TeleCARE system, a multi-agent platform for virtual communities in eldercare.[23] The main idea is to provide a framework upon which a variety of services can be employed. The RoboCare project aims to develop a system consisting of a set of robotic and software agents to control the environment.[24] The system tracks the elderly person with a camera, monitors daily activities and reports on any inconsistency in their execution. Annicchiarico et al. proposed a pedestrian-mobility aid, *iWalker*, aimed to increase elder stability and safety.[25] It tracks a person's intentions in every situation, in the form of agents that deliver monitoring (for example, analysis of gait and activities of daily living), navigation support, and cognitive support (for example, activity reminders). Lakany focused on gait characteristics and designed an algorithm to discriminate between individuals with and without health problems, reflected in any locomotion impairment.[26]

In summary, elder-care systems consist of three main components, as illustrated in Fig. 1. The sensing component is where the sensors are employed, such as an accelerometer, a gyroscope, a camera, and a location system.[8,10,12-15,27,28] The interpretation component detects anomalies such as a fall, an inconsistency in the execution of activities, or a health problem.[8,10,12-15,24],[26,29] The communication component reports the discovered situation to the elder, emergency services, or relatives.[2,21,30,31]

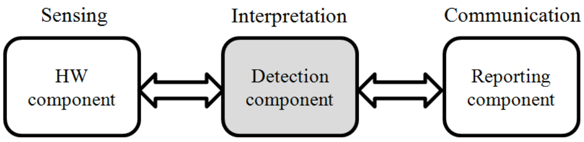


Fig. 1: Abstracted architecture of a typical eldercare system.

This paper focuses on the interpretation component, marked in gray in Fig. 1. We present a multi-agent system consisting of four mechanisms that not only detect falls, but also construct a situational awareness similar to RoboCare.[24] Similar to Lakany and Wu, we also performed an analysis of other potential health problems.[26,29]

From the hardware point of view, our system senses not only accelerations (on which most of the competing systems rely), but also positions of body parts, using the Ubisense system.[32] We made the decision to combine location sensors with inertial sensors for the following reasons: Fall-detection systems based on inertial sensors have high rates of false alarms and fail to recognize specific problems, such as gradually losing consciousness and slowly descending to the floor (as opposed to laying down in bed); accelerometers and gyroscopes do not provide an absolute location, which is essential for high-level awareness, such as recognizing that the person is lying in bed; and cameras may have problems with line of sight and present an unacceptable invasion of privacy. The main drawback of the currently available location systems is their price. However, a significant cost decline is expected in the future, spurred by massive use and advances in hardware.

Compared to the agent-related work, this system is based on a unique, complex, agent architecture. It addresses two goals, each of which is pursued by dedicated agents, but also shares many agents. This makes the architecture efficient and robust. The primary goal is to create a care system able to detect hazardous situations in short-term behavior, such as falls or loss of consciousness. This is one of the most common tasks in eldercare. The second, novel goal is to monitor behavior over longer periods of time and detect changes that indicate decreased performance. Gait disorders, for example, are an important indicator of a variety of health problems, such as joint and muscular diseases, or neurological disorders such as Parkinson's disease.

3. System Architecture

The system is based on the following five requirements:

1. The system is required to monitor a person and detect an emergency situation in real time.
2. The system must be hardware-independent, so it can be easily integrated with various location and inertial sensor systems, and coupled with various schemes for personal interaction.
3. The data representation must allow reasoning about the person in the environment at several abstraction levels. For example, a hazardous fall can be detected with raw sensor data as a sudden change of vertical acceleration, and at the abstract level as a person who quickly lies down.
4. The system must allow for redundancy, which contributes to system stability and robustness and also uses the principle of multiple knowledge. In other words, it combines several methods, sensors, and viewpoints into a single system.
5. The system must be able to provide insight into the person and the environment, as well as an explanation.

A multi-agent architecture meets these system requirements. Each system module, task, or activity is designed as an agent providing a service. Agents are organized into groups at a specific level of abstraction and coordinated by another, higher-level agent. Each agent can be simply modified or replaced, and new or redundant agents can be easily incorporated. An agent provides a service requested by another agent or is triggered by some event. Data is available in the agent's output queue, waiting to be used, requested, or executed by another agent. System knowledge is stored in the system ontology.

3.1. Hierarchical Context-Awareness Architecture

With respect to the described goals and requirements, we present the hierarchical architecture, as shown in Fig. 2.[33] The agents are organized into groups, and the groups are organized into a hierarchical, context awareness that answers the six Ws.[34] *Who* is involved? *Where* did it take place? *When* did it take place? *What* happened? *Why* did it happen? *HoW* did it happen? The *Who* question is trivially solved. The wearable sensors report their identification number, and a specific set of sensors is assigned to a single user. However, multi-user deployment in environments with embedded sensors requires other mechanisms for user identification.

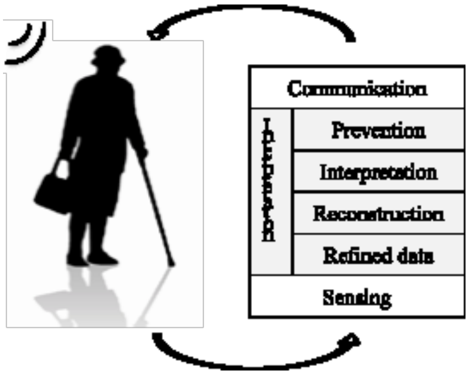


Fig. 2: The hierarchical architecture organizes the agent groups into an increasingly more abstract context awareness and answers the six Ws: Who, Where, What, When, Why and hoW.

The architecture presented in Fig. 2 contains six layers, with a group of agents in each layer.

1. At the lowest layer, an arbitrary location and inertial hardware system is connected to *sensing* agents that send raw data to the second-layer group of agents. These agents answer the *when* and *where* questions.
2. The *refining* agents, a second-layer agent group, represent the first abstraction layer. These agents filter the noise, compute the derived attributes, and map the raw data to the human-body model.
3. The third layer represents the *reconstruction-agent* group. Its main purpose is to answer the *what* question to determine the activity and acceleration of the person in the environment.
4. The fourth level, the *interpretation* group of agents, tries to determine *why*; if a situation is potentially dangerous for the person.
5. The fifth layer is the *prevention-agent* group. Its main task is to answer *how*; to monitor how the person falls, walks, and moves from room to room. Unlike the interpretation group of agents, this group observes longer periods of time and reports any deviations that may indicate a changed psychophysical state or a disease.
6. The last layer is a group of *communication* agents dedicated to personal interaction. These agents alert the person, asking if they are well, phoning relatives or a help center, or graphically displaying the state of the system. One example of the communication agents is an alarm agent that triggers an alarm when a dangerous situation is reported by other agents and deactivates the alarm when the person responds that everything is OK.

3.2. Multi-Agent Architecture

The basic system architecture, presented in Fig. 2, is further elaborated in Fig. 3. Here, the main groups of agents and their interactions are presented. The part of the system on which this paper focuses is presented in the right-hand box in Fig. 3 (interpretation), while the sensing and communication agents are presented in the left-hand boxes (communication and sensing).

The multi-agent architecture design is based on Jade concepts.[35] However, we decided to implement a simplified multi-agent system, which was motivated by increased control over low-level execution, message passing, debugging, and real-time performance. The agents communicate objects such as *measurement*, *snapshot* (a set of measurements), *event*, and *status* by means of messages. These messages are transferred from agent to agent as memory pointers to avoid serialization/de-serialization, and queued in the recipient agent's message queue.

Upon receipt of a new message, each agent updates its internal representation of the environment and communicates its newly perceived environment features to the agents at the higher levels of the context-awareness architecture.

Table 1 presents an overview of the communication among agents, indicated with arrows in Figure 3. An agent in the first column sends the message, while an agent in the first row receives the message. For example, sensor agents send sensor measurements to refining agents, and sensor status to communication agents. They receive enable/disable sensor commands from communication agents.

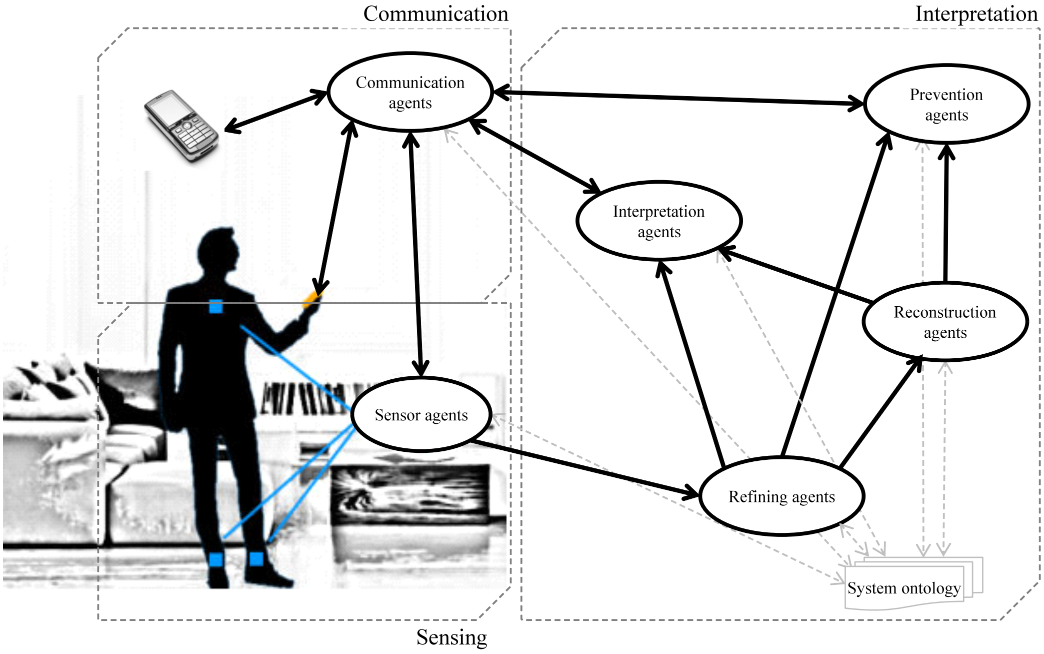


Fig. 3: The agent architecture reveals the various groups of agents. The arrows indicate the communication between agents.

Table 1.Overview of the communication between agent groups.

→	Sensor agents	Refining agents	Reconstruction agents	Interpretation agents	Prevention agents	Communication agents
Sensor agents	-	sensor measurements	-	-	-	sensor status
Refining agents	-	-	filtered data	location, filtered data	location, filtered data	-
Reconstruction agents	-	-	-	activity, additional attributes	activity, additional attributes	-
Interpretation agents	-	-	-	-	-	alarm
Prevention agents	-	-	-	-	-	warning
Communication agents	enable/disable sensor	-	-	alarm response	warning response	-

The described relations between system objects are encoded on an OWL ontology, which is visualized in Figure 4. The boxes marked with letter C represent entity classes corresponding to agent types. Directed arrows between classes denote inheritance, that is, class hierarchy, for example, class *Inertial_sensor* inherits properties from class *Sensor*. The boxes marked with letter P represent properties, that is, relations between classes. For example, class *Sensor* has a property *attachedTo*, which can have a value of class type *Person*. In other words, it specifies that a sensor is attached to a person. The main classes, which are the top nodes in the class hierarchy, are *Sensor, Event, Person, Agent, Classification*, and *Action*. *Agent* class, for example, is inherited by three agent classes: *Sensing, Interpretation* and *Communication*. Each of these classes is then further inherited by specific agent type as described in the following sections.

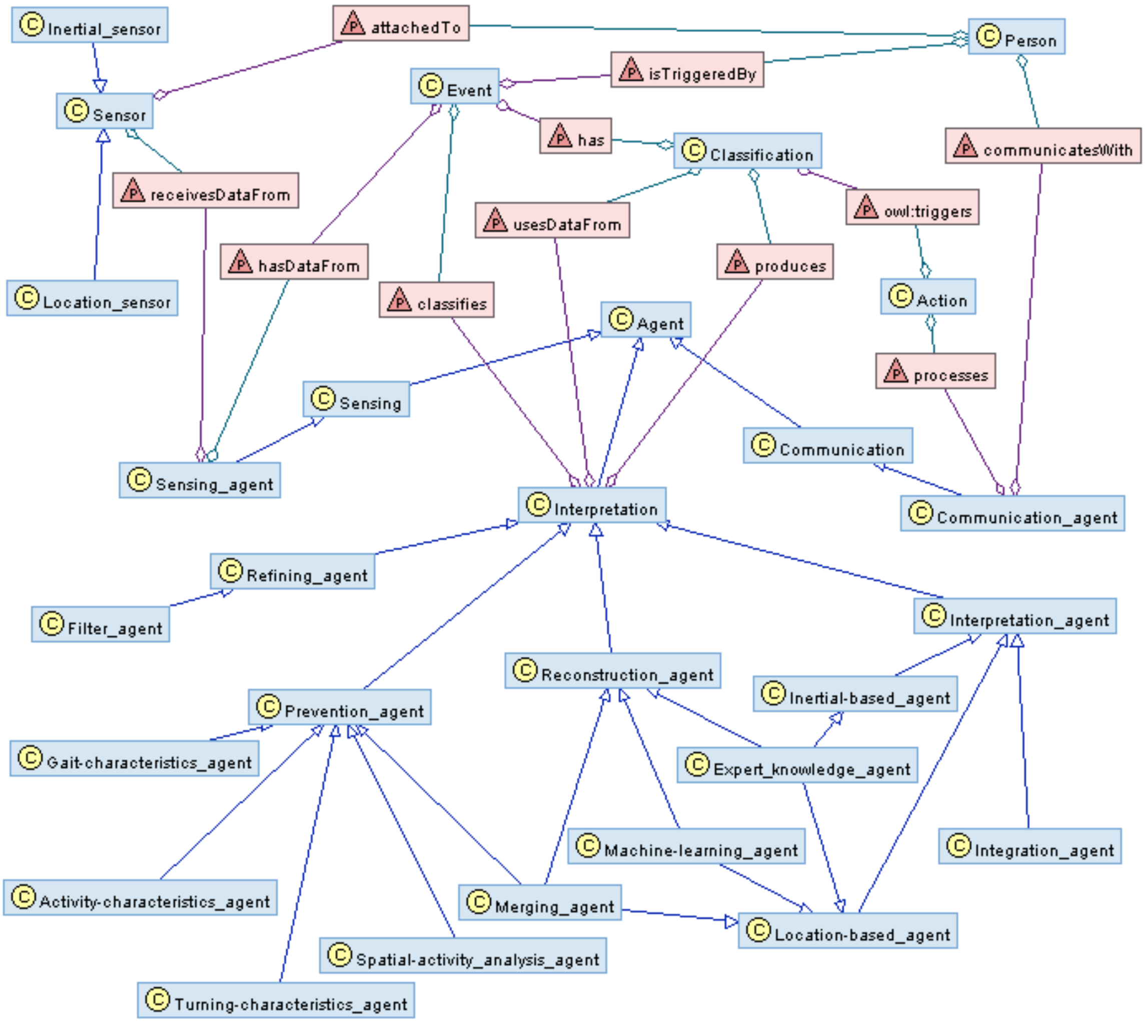


Fig. 4: Relations between the objects in the multi-agent system encoded as an OWL ontology.

The rest of paper focuses on the interpretative component of the system, which consists of four agent groups: Refining, reconstruction, interpretation, and prevention.

4. Agent Groups

4.1. Refining Agents

The refining agents receive measurement messages from the sensing agents. This can be either raw sensor data consisting of body-tag coordinates and accelerations, or other information, such as a “low-battery” status message.

The refining agents’ task is to process the data provided by the sensing agents, filter the noise, compute additional attributes, and map the raw data to a human-body model. If any of the body-tag coordinates is missing, the first step is to estimate the missing coordinates using heuristic rules that compute typical sensor-tag locations, given the activity and locations of other sensor tags.

The refining agents model the human body with rigid-body (non-deformable) components. Rigid-body components are connected to each other with joints to form an articulated body that approximates the human body, as shown in Figure 5. The distance between any two joints is assumed to be constant, regardless of external forces. Blue joints (empty circle) have additional joint constraints such as a minimum/maximum angle. The human-body model is first used to map the measured locations to a valid body configuration and then to compute additional attributes describing body position and movement.

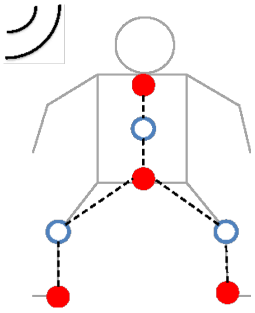


Fig. 5: Human body is modeled with rigid-body components.

In the second refinement step, three filters reduce the considerable noise in measured coordinates.[36] First, a median filter is applied, which eliminates large, short-term changes in signal due to noise. Second, a filter that enforces anatomic constraints is used. It corrects errors, such as an apparent lengthening of a limb. Third, the Kalman filter is applied, which smoothes sharp changes in both location and speed. The effect of the filtering is shown in Figure 6. The blue solid line represents the original signal, while the red dashed line represents the filtered signal in x-direction of a tag attached to the waist.

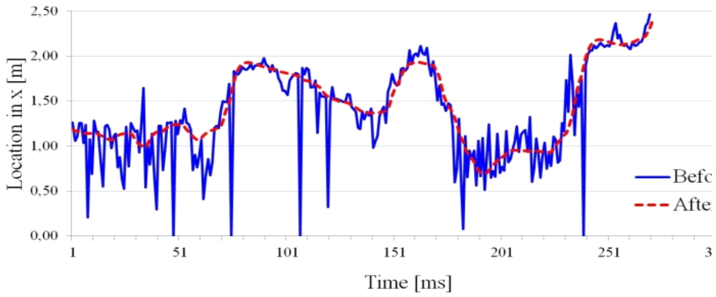


Fig. 6: An example of signal filtered with the three filters.

4.2. *Reconstruction Agents*

The reconstruction agents determine the location and activity of a person at a specific time. These agents provide essential information to activate other agents that assess the situation in the environment and accomplish required actions. The inputs of the agents are the inertial data, filtered location data mapped to the person's body, and the additional attributes describing body position and movement.

These agents' essential task is to recognize a person's specific activity. The standard machine-learning (ML) approach on its own is prone to many difficulties. For example, a classification model trained on a labeled set of data is biased. This means that it can recognize some activities quite reliably while others, especially activities not similar to those in the training set, represent a considerable problem. For example, data of dangerous falls are usually not provided as training examples, since they are hard to obtain. Additionally, although a classifier might be very accurate on average, it may still produce spurious state transitions that do not occur in reality, such as switching from lying to walking in one-tenth of a second.

With respect to the above difficulties we designed some solutions. Instead of using a single learning agent, which might be biased, we use a set of learning agents, each of which provides its own opinion on the classified activity. Instead of a predefined set of ML methods or agents, an arbitrary number of ML and non-ML agents can dynamically cooperate in the classifications. All outputs from the learning agents are merged by the merging agents, which take into account the accuracy of the particular classification agent and the context of the event, and provide a mechanism to decrease the spurious transition effect (explained in Section 4.2.3).

Fig. 7 shows the architecture of the reconstruction-agent group. Each box in the figure represents a group of agents, while the actual number of different implemented agents is presented in the parentheses. There are two groups of learning agents at the lower level: One consisting of expert-knowledge agents and one with ML agents. The classification outputs of the agents at the lower level are merged with the merging agents using heuristics and learning agents to predict and smooth the final classification. We will briefly describe these agents in the following subsections.

4.2.1. Expert-Knowledge Agents

Human activities can be performed in a wide variety of ways. As a consequence, ML models may over-fit to training data, which represents only a small percentage of all the possible activities. The expert-knowledge agents complement the information captured in the training data by adding general knowledge about human activities and the noise of the sensor system(s). For example, an agent might check the input data. If, within two seconds, more than 70 percent of all situations that the system reported the person was laying down, it might decide that the person must be laying down for all of the last two seconds (since nobody lies down 10 times in two seconds).

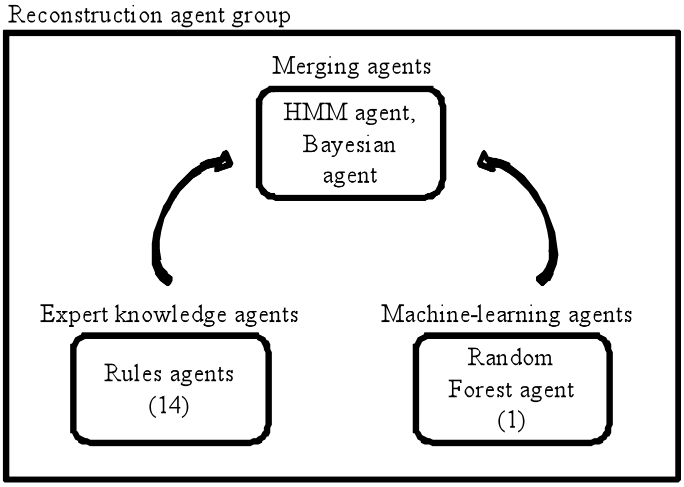


Fig. 7: Agents deployed in the reconstruction-agent group.

The expert-knowledge agents at the first level in Fig. 7 perform activity recognition, according to a set of rules of the form IF *antecedent* THEN *consequent*. They transform observed input sensor data to obtain primitive awareness of the current situation upon which the antecedent is checked. The consequent represents possible activities. Some agents proclaim a single activity and some give multiple possible activities. There are also negation agents, which negate a specific activity or several of them. An example of a rule proclaiming the activity of a laying position is: "If the distance in the z direction between the chest and the ankles of a person is small, and the velocity of the body parts is small, then the person is laying down." An example of an activity negation is: "The person is not sleeping if there were several movements in the last time period." The consequent may also include information about activity dynamics. Differences between standing and walking, are captured in this way. For example, the single activity of walking, may be proclaimed by multiple agents, each of which reasons from a different viewpoint. The antecedent pattern, or the attributes present in the conditions of the antecedent, is determined by human experts.[37] Therefore, the rules are not formally verified. The condition values are set to maximize the accuracy on the training dataset.

4.2.2. Machine-Learning Agents

The machine-learning agent follows the technique we studied in our previous work.[38] It first computes a number of attributes from the sensor values received from the refining agents. The basic attributes use the coordinate system that is fixed with respect to the environment. They consist of the z coordinates, all the tag velocities, the absolute distances and the distances in the z direction between all tag pairs. The *x* and *y* coordinates are omitted because the location where an activity takes place is not important.

The attributes are used to perform classification into one of the predefined activity classes. F_i denotes the feasible sets of features that are computed at a point in time *i*. The attribute vector, which is then used for the classification, composes F_1, F_2, … , F_n successive sets from the time interval 1…*N*. A new attribute vector is then obtained after every update, overlapping with the previous one and providing instant classification for each point in time.

We tested a variety of ML algorithms, including C4.5 decision trees, naïve Bayes, support vector machine, k-NN, bagging, and AdaBoost.[38] Random forest (RF) offered the highest classification accuracy. The most promising ML methods and the attribute sets were implemented in the form of agents. Similar to the expert-knowledge agents, these agents report their classification to the merging agents.

4.2.3. Merging Agents

It might be difficult to form a final opinion when a variety of expert-knowledge and ML agents produce their opinion in the form of an activity (or negation). An elegant way of solving this problem is to combine voting and smoothing mechanisms.

Each vote can be weighted by the classifier's accuracy. When a single classifier mislabels an activity, the other classifiers still have a good chance of providing a correct joint classification. The merging agent dynamically takes into account all the opinions of the lower-level agents, and tracks the accuracy and the context of the learning agents by using a confusion matrix. The initial confusion matrix is obtained with cross-validation on the classifier's training data to estimate the likelihood that the classifier will mislabel an activity or provide the correct activity. Suppose sitting is the actual action. If a classification agent predicts walking, the merging agent examines the confusion matrix to see how likely the classification agent correctly predicts walking, as well as how likely it misclassifies sitting as walking. If the classification agent often misclassifies sitting as walking, while other agents predict sitting, its vote does not count much. More precisely, the merging agent considers each low-level classification as a true activity and computes the likelihood that other classification agents correctly predicted this activity. At the end, the merging agent selects the classification agent's activity that is most probable, and updates the confusion matrix of all the classification agents under the assumption that the selected activity was, in fact, the true activity.

This approach introduces a second level of classification by taking both the classifications and their qualities into account. The merging agent is usually more accurate (by a few percentage points) than any of the individual classification agents. Unlike probabilistic schemes or meta-learning ML systems with predefined combination mechanisms, the merging agent enables dynamic integration of different classification systems.

However, due to the large amount of noise, it still produces some false classifications. In other words, it usually misclassifies single moments or short intervals of an activity. One way to further improve the activity recognition is by taking into account activity continuity. A person's activity cannot switch between walking and sitting every tenth of a second. Transitions between activities that do not occur in reality, but are caused by misclassifications, are considered spurious. We focused on reducing the spurious-activity transitions with hidden Markov models (HMM), realized in the form of agents.[39]

The hidden Markov model is a temporal probabilistic model with two embedded stochastic processes: An unobservable (hidden) process that can only be observed through a second stochastic process that produces the sequence of observations. In contrast to the regular Markov model, the states are not directly visible. Each state has state-transition probabilities (which are visible) and a probability distribution over the possible output symbols. There are two problems that must be solved to use the HMM in practice: How to adjust the model parameters to best describe the sequences of a behavior; and how to choose the corresponding state sequence S that best explains the observations, given the observation sequence s and model λ.[39]

HMM implementation consists of two steps: The learning step, where an agent estimates the parameters of the HMM model; and the classification step, where the agent finds a state sequence that best describes an observation sequence. In the learning step, the agent estimates the parameters by local maximization with the Baum-Welch method, which maximizes the likelihood of the training set.[40] In the classification step, the agent finds the optimal, hidden-state transitions, according to the observation sequence with the Viterbi algorithm.[41] The agent takes an observation sequence and the model as an input and returns the most probable sequence of state transitions. At the end, the states are transformed into the activities.

4.3. Interpretation Agents

The task of the interpretation agents is to understand/explain a given situation and raise an alarm if it is potentially hazardous for the person. The interpretation group of agents constructs the physical awareness of a person in the environment and detects emergency situations caused by a fall or a sudden health problem. These situations are reflected in accelerated movement, or laying or sitting at an inappropriate place (for example, on the ground) for a prolonged period of time. Critical situations are detected based on the history of observed personal activity, which is provided by the reconstruction group of agents.

In a detected critical situation, the interpretation mechanism raises an alarm by requesting service from a communication agent. Each alarm is accompanied by information concerning the alarm type, a description of the critical situation, and a concrete history of events that led up to it. When an alarm is triggered, the agents always receive feedback. The person may confirm, ignore, or reject the alarm. The learning agents in this mechanism then update their knowledge, based on the feedback. If the person does answer at all or confirms the alarm, the agents request communication agents to notify the appropriate caregiver concerning the critical situation. If the person cancels the alarm, this indicates that the person does not need help to cope with the situation. However, the user can always manually trigger an alarm. If the person either rejects the alarm or manually triggers it, the learning agents of this mechanism are adapted to provide more accurate alarms in the future.

The interpretation-agent group presented in Fig. 8 is structured similarly to the reconstruction-agent group. It consists of location- and inertial-based agents. The location-based agents contain the expert-knowledge and learning agents based on ML algorithms. The merging agent makes the final decision of the location-based agents. The inertial-based agents contain threshold-based agents. Integration agents make the final classification of the interpretation agents.

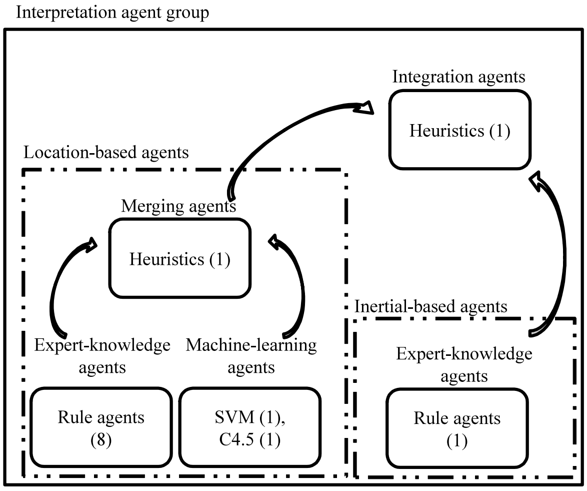


Fig. 8: The interpretation groups of agents.

4.3.1. Location-Based Agents

The location-based agents consist of three groups: Expert-knowledge agents, ML agents, and merging agents, which produce the final decision.

Expert-knowledge agents of the interpretation group contain the expert know-how, similar to the expert-knowledge agents of the reconstruction group, but related to detecting emergency situations that may be caused by a fall or a sudden health problem. The expert-knowledge agents reduce the gap between the real world and the part of it captured in the data available to the ML agents.

Expert-knowledge agents are able to recognize four types of emergency situations: Falling and then laying/sitting immobile at an inappropriate place (that is, place not manually marked during system installation as safe for laying/sitting, such as bed or chair); laying/sitting immobile at an inappropriate place for long period of time; falling and then laying/sitting at an inappropriate place for a long period of time; and laying/sitting at an inappropriate place for a very long period of time. As the input, an agent uses data concerning the person's last fall (detected by the reconstruction agents), the amount of time the person is laying/sitting at the inappropriate place, and the amount of time the person remains immobile. Again, some of the information might be local (stored in the local agent) or global (dynamically gathered from other agents). The detection of each emergency situation is based on the opinions of a set of agents. These agents have predefined forms of rule patterns that are filled by experts, corresponding to the particular needs of the person type. An example of an agent form is: "If a person is laying at an inappropriate place for more than P % in M1 minutes THEN the person is laying down for too long, which indicates that something is wrong," or "if a person is laying in bed for less than M2 minutes, the situation is normal."

ML agents detect emergency situations based on models induced with ML techniques. Unlike expert-knowledge agents, these agents only predict the presence or absence of the emergency situation. The reasoning uses information about the percentage of time the person was involved in specific activities detected by the reconstruction agents, as well as the percentage of time the person was immobile during given time intervals. There are two ML agents, one using a SVM algorithm and the other using a C4.5 algorithm. The other tested ML methods did not perform well for this task.

Merging agent utilizes outputs of the expert-knowledge and the ML agents to decide whether or not the situation is an emergency. The reasoning for this agent is based on a simple heuristic. If both the expert-knowledge and ML agents detect an emergency situation, the merging agent decides that it is an emergency situation. An alarm of the type detected by the expert-knowledge agents (since these agents provide more detailed information about the type of situation) is the output. Furthermore, if only one of these agents detects the emergency situation, this output needs to last more than some predefined time interval to conclude that there is an emergency situation. If the merging agents decide that an emergency situation has occurred, the output is sent to the integration agents.

4.3.2. Inertial-Based Agents

Inertial-based agents contain only the expert-knowledge agents.

Expert-knowledge agents detect emergency situations, such as falls, based on inertial data. A typical inertial pattern during a fall is a decrease in acceleration, followed by an increase (Fig. 9). An accelerometer at rest registers 1 g (the Earth's gravity) and 0 g during free fall. When a person starts falling, the acceleration decreases from 1 g to approximately 0.5 g (perfect free fall is never achieved). Upon impact with the ground, there is a short, strong increase in acceleration.

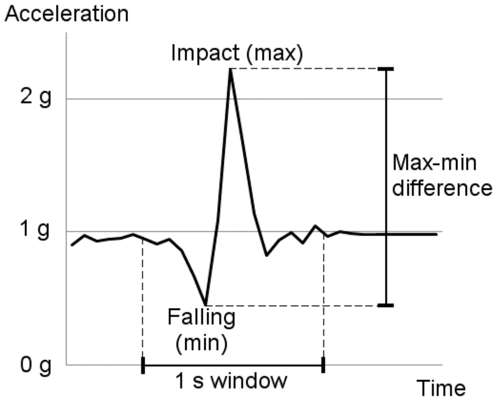


Fig. 9: Acceleration pattern during a fall.

We use the length of the acceleration vector to detect falls, which means that the direction of the acceleration is ignored.[42] The minimum and maximum acceleration within a one-second window are measured. If the difference between the maximum and the minimum exceeds 1 g, and the maximum comes after the minimum, a potential fall has occurred.

The potential fall is confirmed by the measurement of the person's orientation afterwards. We assume that the acceleration vector a = [a_x, a_y, a_z], which consists of the accelerations along the three axes of the sensor, generally points upward (opposite the direction of the Earth's gravity). Z is the axis pointing upward when the person is standing upright. The inclination angle φ between the acceleration vector and the z axis thus indicates the person's orientation, computed as follows:

$$\cos\varphi = \frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}}. \tag{1}$$

A person is oriented upright if –30° < φ < 30°. If acceleration that exceeds the threshold is detected (as described previously), and the orientation afterward is not upright, a fall has occurred.

Each person has a characteristic, upright torso inclination during static upright activities (such as standing). The chest and waist inertial sensors may register different inclination angles during these upright activities. Because of this, the extracted inclination angles during these activities should be adjusted. The average orientation during 15 seconds of standing (upright activity) is measured as φ_0. Subsequently, when a new upright orientation φ_{raw} is measured, the orientation is adapted as follows:

$$\varphi = \varphi_{\text{raw}} - \varphi_0. \tag{2}$$

This way, normalization between different individuals' sensor inclination angles is achieved. If the potential fall is confirmed, the threshold-based agent outputs an alarm to the integration agents.

4.3.3. Integration Agent

The purpose of the integration agent is to provide the final judgment about whether or not there is an emergency situation. This module uses the output of the location- and inertial-based agents as its input. The reasoning for this agent is based on a simple heuristic using context-awareness, where the context is the location of the person. If both the location- and inertial-based agents detect an emergency situation, then the final decision is that there is an emergency situation. Furthermore, if only one of these agents detect the emergency situation, the alarm output is generated, according to predefined rules. The alarm is raised when:

- The location-based agents detect a fall AND the person is not moving afterward; OR
- The inertial-based agents detect a fall AND the location is not intended for laying down.

4.4. Prevention Agents

This section presents the agent group for recognizing behavioral changes that could indicate the beginning of a disease. It only provides a warning to caregivers. A set of agents observe the person's behavior, each collecting a specific subset of the person's behavioral data. These agents automatically build behavioral models that are constantly updated. After predefined time periods, they classify the current behavior with these models to recognize behavioral changes. Since each agent only partially observes the person's behavior, a merging agent collects their observations and merges them into the final behavioral observation. These agents are described in detail in the following sections and presented in Fig. 10.

4.4.1. Gait-Characteristics Agents

These agents focus on a person's gait to recognize any changes. Data is first collected while the person is walking. An agent then extracts steps from the walking and calculates significant attributes for each step. These attributes are then averaged over a given time period, thus forming a walking signature. Finally, the walking signature is compared to previous signatures. The walk is marked as potentially risky if it significantly deviates.

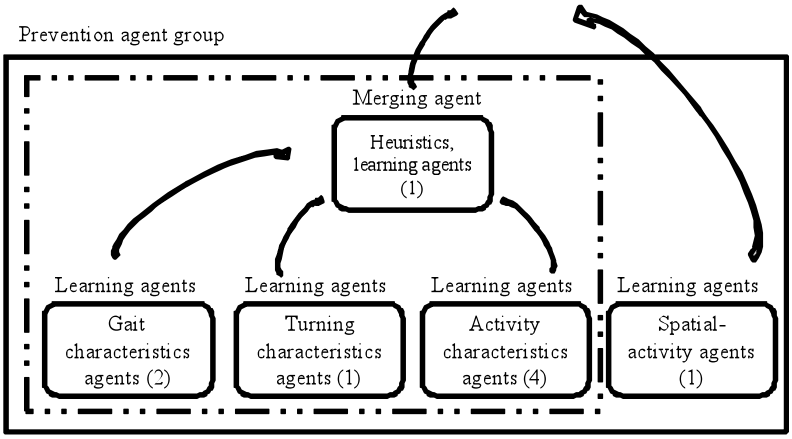

Prevention agent group
Merging agent
Heuristics, learning agents (1)
Learning agents
Gait characteristics agents (2)
Learning agents
Turning characteristics agents (1)
Learning agents
Activity characteristics agents (4)
Learning agents
Spatial-activity agents (1)

Fig.10: Architecture of the prevention-agent group.

The step agent extracts the steps by observing the ankle's coordinates. Each step begins when the ankle starts to move, or when the z coordinate starts to rise, and the x and y coordinates start to change. It ends with the foot standing still on the floor, or when the x and y coordinates do not change. Each step is a union of the steps with the left and the right foot. Next, a set of attributes is calculated for each step, based on medical literature.[43-46] These attributes include support (foot on the ground), swing (foot off the ground) and step (support + swing) times, double support time (both feet on the ground), step length and width, and maximum distance of the foot from the ground.

The walking-signature deviation is recognized by another agent that performs a Local Outlier Factor (LOF) algorithm, which measures how much the walking signature differs from its previous, usual values or, in other words, to what degree the signature is an outlier.[47]

4.4.2. Turning-Characteristics Agent

Significant walking characteristics are also observed during a turn, in addition to gait. The first step is the turning recognition. This is done by an agent that tracks the person's walking line. Turning occurs when the person does not go straight, but turns in the same direction, left or right, for a predefined time period and number of steps. When the turning is concluded, the following attributes are computed: speed while turning, angular velocity, and turn radius. The usual turning characteristics and the deviation detection are then utilized by a learning agent based on the LOF algorithm.

4.4.3. Activity-Characteristics Agent

Although walking is the most active, important activity to assess the person's behavior, these agents are devoted to observing other activities that also contribute to detecting behavioral changes. The agents are not focused on a specific activity but observe general behavior during several activities. To achieve the requested generality, a set of simple attributes is used, containing only the activity time and the average speed of all available locations of the body parts, such as the general speed during walking and laying, the walking speed, and the time of activity transitions during sitting down and standing. These attributes are averaged over a given time period and used by a learning agent to detect deviations with the LOF algorithm (similar to the walking-signature and turning agents).

4.4.4. Spatial-Activity Analysis Agent

The spatial-activity agent is focused on a person's daily behavior. It observes a person over a longer period of time, such as dynamics in their apartment. It does this by monitoring activities performed during the day, combined with spatial information. The agent collects a variety of statistics, organized in a spatial-activity matrix that is divided into four sections. These sections are the spatial-spatial part, describing the share of time spent in particular areas of the room and the transition distribution between different areas: The activity-activity part, which includes the shares of time spent performing a particular activity and the transition distribution between activities; the spatial-activity part, describing distribution of activities over parts of the room; and the activity-spatial part, describing the distribution of parts of the room over activities. The spatial-activity matrix is computed for each day, capturing behavioral dynamics during a day. An example matrix with three activities and four rooms is shown in Fig. 11.

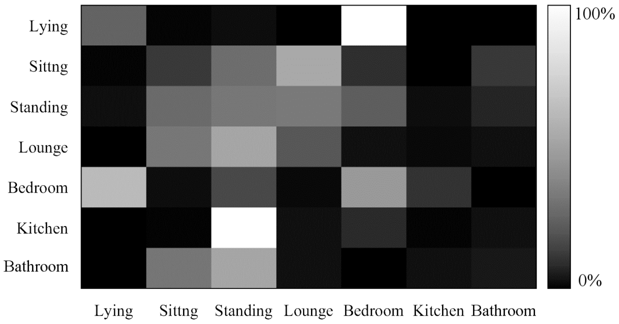


Fig. 11: Spatial-activity matrix for the detection of unusual long-term movement.

The spatial-activity matrix for each day is unrolled into a vector, after which the Principal Component Analysis is performed to reduce its dimensionality. Such vectors are stored for an initial training period, during which the movement is considered normal. These vectors represent the training data. Afterwards, the LOF algorithm is used to detect daily behavior that deviates from the training data[48].

4.4.5. Merging Agent

This agent receives messages from the lower-level agents of the prevention group and integrates them into the final prevention message. Each prevention agent asynchronously sends the messages to the integration agent, which collects the messages within a predefined period and analyses their outlier degrees. If a single outlier degree is higher than the higher warning bound, or several outlier degrees surpass the lower warning bound, the prevention agent issues a warning. High and low warning bounds are experimentally set for each individual agent.

5. Prototype Implementation

For the prototype deployment, we organized a room as an apartment, with an area of about 25 square meters. The testing room was equipped with a bed and a few chairs and tables, and was divided into six areas: A kitchen; a sleeping area; a living room; a toilet; and a corridor for entering and leaving the room.

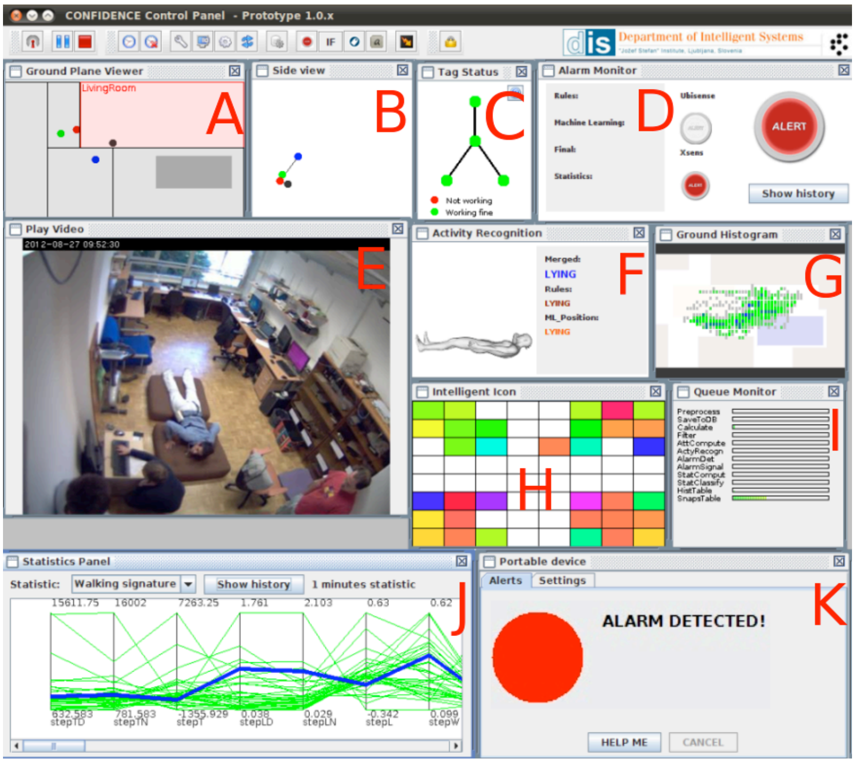
We selected a commercially available location system, Ubisense, and an inertial system, Xsens, for the sensing component.[32,49] Ubisense allows local positioning by tracking a set of reasonably small tags attached to a person. A sampling frequency of approximately 10 Hz can be achieved with no more than four tags attached to a person. Each tag maintains radio contact with a sensor mounted on the wall. These sensors use ultra-wideband

(UWB) technology to detect and react to the position, and make use of both the time difference of arrival and the angle of arrival to calculate the location. In a typical open environment, a location accuracy of about 15 cm can be achieved across 95 percent of the readings.[49] However, the accuracy occasionally drops below 200 cm in real-life scenarios, which represents a challenge for preprocessing and filtering, as described in Section 4.1.

Luštrek et al. studied how tag placement on the body affects classification accuracy of various agents.[51] In general, more tags enable more accurate classification. However, due to high noise, more tags do not necessarily improve the results. For example, the accuracy of the activity-recognition agents in our experiments was comparable when using only four or eight tags. However, fewer than two tags significantly affected classification accuracy. Due to our results, and that the Ubisense sampling rate of 10 Hz is limited to up to four tags, we decided to position the tags at the following locations: Chest, belt (optional), and left and right ankles.

We developed a set of graphical and communication agents as a communications component: An agent to inform the person about an alarm or warning; an agent that interacts with the person (to raise, cancel, or hold an alarm); graphical agents that explain the current state of the particular agent; and agents that explain why a specific alarm or warning was raised. Our prototype implementation is currently PC-based while we are in the process of extending it with more advanced, service-based agents, such as an emergency call service, a portable device for alarm inhibition, a mobile phone, and a remote web-based monitoring application.

The communication and graphical agents enable insight into the multi-agent environment by presenting the state of a particular agent on a GUI, as depicted in Fig. 12. The personal interface consists of 11 frames, marked with letters. Three frames display the state of the reconstruction-agent group: Top view of a person in the room (frame A); side view as coordinates x and z (frame B); and the recognized activity by particular agents and joint classification (frame F). The state of the interpretation-agent group is displayed in frame D, with decisions of particular agents and the final decision. Frame E is dedicated to a live video stream. It is only used for developing and testing purposes, and is not part of the final system. The prevention-agent group is displayed in the following frames: Histogram of the person's presence at a specific location in the room (frame G), a visualization of the spatial-activity analysis agent (frame H), and the state of other prevention agents (frame J), using parallel coordinates, which enable a clear distinction between the normal and abnormal events. The status of the hardware agents and the available tags is presented in frame C, while the person-interaction agent (to cancel or manually trigger an alarm) is in frame K. The other frames are relevant for the developers.


CONFIDENCE Control Panel - Prototype 1.0.x

Merged: LYING
Rules: LYING
ML_Position: LYING

Statistic: Walking signature
1 minutes statistic

ALARM DETECTED!

Fig. 12: The communication and graphical interface of the prototype system for eldercare.

We designed two sets of experiments showing the capabilities of the interpretation and prevention group of agents. The first experiment is devoted to fall detection, where we present complex situations that can be easily misinterpreted by an acceleration-based fall detector. The second experiment verifies how the prevention group of agents adapts to a person and detects disabilities. The experiments were supervised by a medical doctor to ensure that the imitated movement, falls, and symptoms were realistic.

5.1. Fall-Detection Experiments

The first experiment explores the fall-detection capabilities of the interpretation-agent group. For this purpose, we designed a complex scenario, including three situations when an alarm must be raised and two situations where no alarm should be reported. The situations in which the alarm must be raised are:

1. Tripping: A typical fall that occurs quickly and ends with a strong impact.
2. Falling slowly: May occur when a person becomes weak and slowly collapses without a strong impact (slow fall, potentially not detected by inertial sensors).
3. Tripping + standing up: Occurs if the person falls, but is not injured enough to be unable to stand up. However, it is still treated as a fall, because it is not uncommon for an elder to suffer an injury and either not realize it, or not realize its seriousness (potentially not detected by location sensors).

Situations that might lead to a false alarm, but are actually safe, are:

1. Laying down quickly in bed: This is not a fall, but may appear like one to inertial sensors.
2. Searching for an object on the ground, either on all fours or laying down: This is also not a fall, but may appear like one to location sensors.

These two sets of situations reveal the interpretation-agent group's ability to carry out context-dependent reasoning, or distinguish between an alarm situation and normal behavior consisting of the same activities, based on their context. For example, searching under the table/bed may be falsely perceived as laying on the ground, since the person is close to the ground. However, it differs from laying on the ground by the amount of bodily movement and the length of time the person is on the ground.

The prototype was tested on recordings of 10 young people, six males and four females, each performing the above-mentioned test scenario five times. There were a total of 50 recordings. The recordings were made in the experimental room described above. None of the test recordings were used to train or tune any of the methods, and none of the volunteers participated in any of the training recordings. The ML agents were trained on similar recordings of five young people, three males and two females (available from the UCI repository).[53] The results of the tests are shown in Table 2.

Table 2. Accuracy of fall detection.

Event	Location-based agents		Inertial-based agents	Integration agents
	4 tags	1 tag		
		Falls		
1. Tripping	100.0%	93.9%	100.0%	100.0%
2. Falling slowly	95.9%	100.0%	10.6%	100.0%
3. Tripping + standing up	0.0%	0.0%	100.0%	100.0%
		Non-falls		
4. Laying down quickly	100.0%	100.0%	34.0%	100.0%
5. Searching on the ground	83.7%	61.2%	100.0%	61.2%
Average	77.5%	70.9%	68.9%	92.2%

The first two columns show the accuracy of fall detection with just location-based agents (Section 4.3.1), either with four tags or with one chest tag. The next column shows the accuracy of fall detection using just the inertial-based agents (Section 4.3.2). The last column shows the final decision using the integration agents (Section 4.3.3).

Table 2 shows that each agent type has a limited view of an event and therefore fails to correctly recognize some of the events as falls or non-falls. However, since the sensors complement each other, using integration agents yielded almost perfect fall detection. One could test the fall detection on additional events in which even the integration agents would fail (for example, fainting on the bed). However, our purpose was not to show that the system can detect all fall types, but rather that combining location- and inertial-based agents can significantly improve fall detection over what can be achieved with either sensor type on its own.

Looking at the individual fall types, tripping is indeed a typical fall, which was accurately recognized by both types of agents. Slowly falling was easy for the location-based agents to recognize, since they rely on the recognition of laying down. However, from the inertial-based agents' viewpoint it appeared as though laying down was voluntary. Tripping + standing up was impossible for the location-based agents to recognize, because the period of laying down was too short. However, the inertial-based agents recognized it perfectly, since there was a strong impact and some laying down afterward.

Of the non-fall events, the location-based agents perfectly recognized quickly laying down, because they could use the information about the bed and considered laying there safe. However, laying down quickly was almost indistinguishable from a fall for the inertial-based agents. Searching on the ground was somewhat difficult for the location-based agents to recognize, since it involved laying at a location not intended for laying, similar to a fall. However, the inertial-based agents performed perfectly, since there was no strong impact. The integration agents did not always perform perfectly, since they depended on the amount of laying on the floor and moving while searching.

In addition to the presented experiment, we also followed Gjoreski et al.'s suggestions for the best location of inertial-sensor placement for detecting falls.[52] The best placement was on the chest when using both sensor types.

5.2. Health-Deterioration Detection Experiments

The second experiment verified the prevention agents' ability to detect changes in behavior, indicating current or emerging health deterioration or disease. Testing the health-deterioration detection would ideally require many days of recording daily activities. To speed up the experiments, we condensed four days of activities into recordings that lasted around 25 minutes each. The result was two normal days and two abnormal days: One day of slower movement with more rest and one day of limping. The series of activities for a normal and an abnormal day are shown in Table 3. The experiment, supervised by a medical doctor, was performed by two healthy volunteers imitating a health problem.

The detection of unusual, short-term movement was performed in five-minute intervals. One normal day was used for training to set the warning bounds. After the bounds were set, the method was tested on the remaining normal day and the two abnormal days. No unusual movement was detected on the normal day. Table 4 shows the detections of unusual movement on the abnormal days as shaded cells for each interval and attribute group (groups described in Sections 4.4.1–4.4.3). At least one attribute group detected unusual movement during each interval, and usually more than one. This is sufficient, since it is not necessary that all attribute groups detect unusual movement to realize that a person is moving unusually. It is not even essential that such movement is detected during each interval, since it is usually not a sign of an immediate threat to the person's health. This gives a margin of safety in case the performance is worse in more realistic settings.

Table 3. Results for detection of unusual short-term movement

Day part	First normal day	Slow day
Pre-breakfast	Sleep Visit bathroom	Sleep Visit bathroom
Breakfast	Prepare breakfast Eat breakfast Clean up after breakfast Visit bathroom	Prepare breakfast Eat breakfast Rest by the table Visit bathroom Clean up after breakfast
Morning	Read newspaper Go out	Read in the bed Household chores Rest in a chair
Lunch	Prepare lunch Eat lunch Visit bathroom Clean up after lunch	Prepare lunch Eat lunch Visit bathroom Rest in the bed Clean up after lunch
Afternoon	Play solitaire Household chores Read	Watch TV
Dinner	Prepare dinner Eat dinner Visit bathroom Clean up after dinner	Eat a snack Visit bathroom
Post-dinner	Watch TV Sleep	Read in the bed Sleep

We recorded additional data for detecting unusual, long-term movement. It was tested on complete days, using the scenarios in Table 3. Table 5 shows the results, in terms of the LOF values (obtained with leave-one-out validation) representing the degree of unusualness of the movement. The LOF values during the abnormal days are much higher, making them easy to distinguish from the normal days.

6. Discussion

6.1. Deployment Aspects and Social Acceptability

The deployment of such a system includes location and inertial systems (Ubisense and/or Xsens) and a processing computer. The location system requires installation of anchor points and a calibration procedure in which the base coordinate system is established. The inertial system does not require any calibration procedure.

Table 4. Results of the detection of unusual movement. The outlier detections are marked in gray.

	Gait characteristics		*Turning characteristics*	*Activity characteristics*			
Day, interval	Walking signature	Walking speed	Turning	Sitting/laying down	Standing up	General speed upright	General speed laying
Slow 1	■			■			
Slow 2		■	■			■	
Slow 3		■				■	■
Slow 4		■					
Slow 5		■			■		
Limping 1				■	■		■
Limping 2					■		■
Limping 3							■
Limping 4		■	■			■	
Limping 5	■			■			

Table 5. Results of the detection of unusual long-term movement.

Day	First person	Second person
First normal day	0.619	0.887
Second normal day	0.694	0.904
Third normal day	0.652	0.843
Forth normal day	0.601	0.832
Slow day	3.274	5.451
Limping day	2.369	4.519

The proposed multi-agent system (source code available from authors on request[53]) is installed on a processing computer that runs Java and a relational database, such as PostgreSQL. The system configuration includes communication parameters for the sensing hardware, sensor positions on the body, and specification of areas safe for laying down. This can be configured using the installation wizard, which is run before the system is used for the first time, for example, have a person show how they usually perform activities, specify their height to refine the reconstruction-agent mechanisms, set parameters that describe their normal-day dynamics to relax or strengthen the alert sensitivity, and set privacy-related services for reporting and monitoring. The recorded activities represent a supervised dataset for a specific end-user and adapt parameters and models of interpretation agents to that user.[55]

The Ubisense location tags' battery life is several weeks, while the XSens inertial sensor's battery life is approximately one day. From a practical point of view, the most convenient sensor configuration for the user is a single location tag, but overall system performance is affected (Table 3). Combining a location tag with an inertial sensor in a single enclosure offers excellent performance and reasonable convenience. Only one enclosure needs to be attached to the body, although the battery must be charged daily.

The system was subjected to end-user testing by elderly people, within the scope of the Confidence project. The results reported that users perceived it as a tool to improve their sense of security, and appreciated the possibility of monitoring health parameters, as well as raising a warning in case of deterioration in health.[56] Concerns were mostly focused on hardware experience, such as stigmatization caused by sensors visibility, sensor battery autonomy, and cables and other hardware required to install system. However, these issues are largely related to the current prototype status and most could easily be addressed in a commercial product.

6.2. Limitations of the Study

The reported experiments were performed in a controlled environment with young volunteers. Experiments, such as fall detection, are too dangerous for elderly subjects to perform, therefore, no elderly was involved in this study; however, we introduced several measures to ensure the quality of the experiments, for example, all the scenarios were designed in collaboration with a medical doctor. Since the experiments were performed in a controlled environment, that is, a laboratory, deployment in a real apartment may face certain issues related to hardware reliability and, consequently, system performance.

7. Conclusions

This paper presented a multi-agent system that integrates four major groups of intelligent agents: Noise suppression, reconstruction, interpretation, and prevention. The first group obtains the sensor data and attenuates the noise. The second group provides the person's location and activity. The third group understands and reasons about the reconstructed situation to dynamically react to exigent circumstances. The fourth group detects deviations in the behavior of an elderly person that might indicate health deterioration or disease. The agents are arranged horizontally, to contribute various interpretations of the situation, and vertically, to provide increasingly more abstract situational awareness. Each agent has its own strong and weak points. However, advanced combination and integration overcome the individual weaknesses and combine different aspects into a reliable interpretation.

The results from the fall-detection experiment show that context-dependent reasoning can detect complex scenarios that might be misinterpreted by acceleration-based systems. The preliminary results for the detection of health deterioration are encouraging, showing a potential for early discovery of a potential health problem that could be perilous. This is probably the most important part of the system, setting it apart from similar systems that already tackle falls and other clear-cut problems. Overall, hundreds of hours of tests were performed including a conference demo at AAMAS and an on-line presentation to EU reviewers, with 100-percent performance for scenarios prepared in advance and defined on the fly.[53]

To increase the usability, performance, and effectiveness of the interpretative multi-agent system, it can be adapted to an end-user. While the initialization wizard optimizes the initial parameters of the system to a particular person, we have also developed additional procedures for personal adaptation during the system operation, in particular, dynamic adaptation to false and undiscovered alarms.[57]

Future work is mainly focused on improving the reconstruction agents and devising effective and pro-active agents to broaden the scope of the awareness and prevention mechanism. We plan to incorporate the temporal sequence of daily activities into the prevention agents and improve the output merging of the individual agents. Ultimately, the success of the interpretative mechanism's ambitious goals strongly depends on the ability to suit and adapt to an individual.

Hopefully, the proposed multi-agent system will provide novel perspectives for the interpretative components of care systems, as well as a number of technically interesting solutions for reconstruction, awareness, and prevention mechanisms.

Acknowledgments

This work was partly supported by the Slovenian Research Agency under the Research Programme P2-0209 Artificial Intelligence and Intelligent Systems, and partly from the European Community's Framework Programme FP7/2007–2013 under grant agreement No. 214986. The work of Violeta Mirchevska was partly financed by the European Union, European Social Found.

References

1. European Commission, *Demography Report 2010* (Publications Office of the European Union, Luxembourg, 2011).
2. R. Means, S. Richards, R. Smith, *Community Care: Policy and Practice* (Palgrave, MacMillan, 2008).
3. *Confidence* (2012), http://www.confidence-eu.org.
4. M. Gams, *Weak Intelligence: Through the Principle and Paradox of Multiple Knowledge* (Nova Science Publishers, 2001).
5. L. Z. Rubenstein, K. R. Josephson, A. S. Robbins, Falls in the nursing home, *Annals of Internal Medicine* **121** (1994), pp. 442–51.
6. G. F. Fuller, Falls in the elderly, *American Family Physician*, **61** (2000), pp. 2159–68.

7. J. I. Pan, C. J. Yung, C. C. Liang, L. F. Lai, An intelligent homecare emergency service system for elder falling, *World Congress on Medical Physics and Biomedical Engineering* (Springer, Berlin Heidelberg, 2006), pp. 424–8.

8. A. K. Bourke, G. M. Lyons, A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor, *Medical Engineering &Physics*, **30** (2008), 84–90.

9. K. Doughty, R. Lewis, A. McIntosh, The design of a practical and reliable fall detector for community and institutional telecare, *Journal of Telemedicine and Telecare*, **6** (2000), 150–4.

10. M. Kangas, A. Konttila, I. Winblad, T. Jamsa, Determination of simple thresholds for accelerometry-based parameters for fall detection, *Proceedings of the 29th Annual International Conference of the IEEE*, Engineering in Medicine and Biology Society (IEEE, 2007), pp. 1367–70.

11. D. J. Willis, *Ambulation Monitoring and Fall Detection System using Dynamic Belief Networks*. PhD Thesis, School of Computer Science and Software Engineering (Monash University, 2000).

12. T. Zhang, J. Wang, J. Liu, P. Hou, Fall detection by wearable sensor and one-class SVM algorithm, *Lecture Notes in Control and Information Science* (Springer, Berlin Heidelberg, 2006), pp. 858–63.

13. Z. Fu, E. Culurciello, P. Lichtsteiner, T. Delbruck, Fall detection using an address-event temporal contrast vision sensor, *Proceedings of the IEEE International Symposium on Circuits and Systems – ISCAS 2008* (IEEE, 2008), pp. 424–7.

14. *AlertOne Services, Inc. iLife™ Fall Detection Sensor* (2012), http://www.falldetection.com.

15. *Zenio, Zenio Fall Detector* (2012), http://www.zenio.be/product/8.html.

16. Q. Li, J. A. Stankovic, M.A. Hanson, et al., Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information, *Proceedings of the Sixth International Workshop on Wearable and Implantable Body Sensor Networks* (IEEE, Berkeley, 2009), pp. 138–43.

17. D. Isern, D. Sanchez, A. Moreno, Agents applied in health care: A review, *International Journal of Medical Informatics*, **79** (2010), 145–66.

18. T. A. Illarramendi, M. I. Bagues, J. Bermudez, A. Goni, Aingeru: An innovating system for tele-assistance of elderly people, *Journal on Information Technology in Healthcare* **2** (2004), pp. 205–14.

19. V. Koutkias, I. Chouvarda, N. Maglaveras, A multi-agent system enhancing home-care health services for chronic disease management, *IEEE Transactions on Information Technology in Biomedicine* **9** (2005), 528–37.

20. L. Cervantes, Y. S. Lee, H. Yang, et al., Agent-based intelligent decision support for the home healthcare environment, *Proceedings of the 1st International Conference on Advances in Hybrid Information Technology*, in *Lecture Notes in Control and Information Science* (Springer, Berlin Heidelberg, 2007), pp. 414–424.

21. A. Moreno, A. Valls, D. Riaño. PalliaSys: Agent-based proactive monitoring of palliative patients, *IV International Workshop on Practical Applications of Agents and Multi-Agent Systems*, pp. 101–10.

22. A. Moreno, A. Valls, D. Riaño. Agent-based alarm management in a palliative care unit, *III Workshop on Agents Applied in Health Care, 19th International Joint Conference on Artificial Intelligence (2005)*, pp. 6–60.

23. L. M. Camarinha-Matos, H. Afsarmanesh, A multi-agent based infrastructure to support virtual communities in elderly care, *International Journal of Networking and Virtual Organisations*, **2** (2004), 246–66.

24. A. Cesta, F. Pecora, Integrating intelligent systems for elder care in RoboCare, W. C. Mann and A. Helal (Eds.), *Promoting Independence for Older Persons with Disabilities* (IOS Press, 2006), pp. 65–73.

25. R. Annicchiarico, C. Barrué, T. Benedico, F. et al., The i-Walker: An intelligent pedestrian mobility aid, *Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence* (IOS Press, 2008), pp. 708–12.

26. H. Lakany, Extracting a diagnostic gait signature, *Pattern Recognition*, **41** (2008), 1644–54.

27. B. Kaluža, V. Mirchevska, E. Dovgan, et al., An agent-based approach to care in independent living, *AmI '10: Proceedings of the European Conference on Ambient Intelligence, Lecture Notes in Computer Science*, **6439** (2010), pp. 177–186.

28. G. Sukthankar, K. Sycara, A cost minimization approach to human behavior recognition, *AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-agent Systems* (ACM, New York, 2005), pp. 1067–74.

29. G. Wu, Distinguishing fall activities from normal activities by velocity characteristics, *Journal of Biomechanics*, **33** (2000), 1497–1500.

30. G. Fiol-Roig, D. Arellano, F. J. P. Lopez, et al., The intelligent butler: A virtual agent for disabled and elderly people assistance, *Advances in Soft Computing* (Springer-Verlag, Berlin Heidelberg, 2008), pp. 375–84.

31. J. I. Pan, C. J. Yung, C. C. Liang, L. F. Lai, An intelligent homecare emergency service system for elder falling, *World Congress on Medical Physics and Biomedical Engineering* (Springer-Verlag, Berlin Heidelberg, 2006), pp. 424–8.

32. *Ubisense location system*, http://www.ubisense.net.

33. R. A. Brooks, Intelligence without representation, *Artificial Intelligence*, **47** (1991), 139–59.

34. J. C. Augusto, Ambient intelligence: The confluence of ubiquitous/pervasive computing and artificial intelligence, *Intelligent Computing Everywhere* (Springer-Verlag, London, 2007).

35. Jade, http://jade.tilab.com.

36. B. Kaluža, E. Dovgan, Glajenje trajektorij gibanja človeškega telesa zajetih z radijsko tehnologijo, *Proceedings of the 13th International Multiconference Information Society* (IS, Ljubljana, 2009), pp. 97–100.

37. V. Mirchevska, M. Luštrek, M. Gams, Combining machine learning and expert knowledge for classifying human posture, *Proceedings of the Eighteenth International Electrotechnical and Computer Science Conference* (ERK, Slovenia, 2009), pp. 183–6.

38. M. Luštrek, B. Kaluža, Fall detection and activity recognition with machine learning, *Informatica* **33** (2009), 197–204.

39. L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Readings in Speech Recognition* (Morgan Kaufmann, San Francisco, 1990).

40. L. E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics*, **41** (1970).

41. A. J. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *Information Theory, IEEE Transactionson*, **13** (1967).

42. H. Gjoreski, M. Luštrek, M. Gams, Accelerometer placement for posture recognition and fall detection, *Proceedings IE* (IEEE, 2011), pp. 47–54.

43. T. L. Heiden, D. J. Sanderson, J. T. Inglis, G. P. Siegmund, Adaptations to normal human gait on potentially slippery surfaces: The effects of awareness and prior slip experience, *Gait & Posture*, **24** (2006).

44. D. S. Marigold, A. E. Oatla, Age-related changes in gait for multi-surface terrain, *Gait & Posture*, **27** (2008).

45. R. Paróczai, Z. Bejek, A. Illyés, et al., Gait parameters of healthy, elderly people, *Facta Universitatis*, **4** (2006).

46. R.C. Woledge, D. B. Birtles, D. J. Newham, The variable component of lateral body sway during walking in young and older humans, *Journal of Gerontology: Medical Sciences 60A*, **11** (2005).

47. M. M. Breunig, H. P. Kriegel, R. T. Ng, J. Sander, LOF: Identifying density-based local outliers, *Proceedings SIGMOD '00* (ACM, 2000), pp. 93–104.

48. B. Kaluža, M. Gams, Analysis of Daily-Living Dynamics, *Journal of Ambient Intelligence and Smart Environments*, **4** (2012), pp. 403–413.

49. XSens Inertial Motion Tracker (2012) http://www.xsens.com.

50. P. Steggles, S. Gschwind, *Ubisense-A Smart Space Platform: Technical Report* (Ubisense, 2005).

51. M. Luštrek, B. Kaluža, E. Dovgan, et al., Behavior analysis based on coordinates of body tags, *AmI '09: Proceedings of the European Conference on Ambient Intelligence, Lecture Notes in Computer Science*, **5859** (2009), pp. 14–23.

52. H. Gjoreski, M. Luštrek, M. Gams, Context-based fall detection using inertial and location sensors, *Proceedings of the International Joint Conference on Ambient Intelligence* (*Lecture notes in computer science*, 2012), pp. 1–16.

53. Confidence project website (2012) http://dis.ijs.si/confidence.

54. Luštrek, M., Kaluža, B., Piltaver, R., Krivec, J., and Vidulin, V. 2010. Localization Data for Person Activity Data Set. http://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity.

55. B. Cvetković, B. Kaluža, M. Luštrek, and M. Gams, Mutliclassifier adaptive training: Specializing an activity-recognition classifier to using semi-supervised learning, *International Joint Conference on Ambient Intelligence 2012, Lecture Notes in Computer Science*, **7683**, pp. 193–207 .

56. Igone Velez et al., Deliverable D8.5: Final report, Confidence Project, 2011.

57. V. Mirchevska, M. Luštrek, and M. Gams, Combining Domain Knowledge and Machine Learning for Robust Fall Detection, *Expert Systems* (2013), preprint published online.