

Towards a social and context-aware multi-sensor fall detection and risk assessment platform

F. De Backere^{a,*}, F. Ongenae^a, F. Van den Abeele^a, J. Nelis^a, P. Bonte^a,
E. Clement^b, M. Philpott^b, J. Hoebeke^a, S. Verstichel^a, A. Ackaert^a,
F. De Turck^a

^a*Department of Information Technology (INTEC), Ghent University – iMinds,
Gaston Crommenlaan 8, bus 201, B-9050 Gent, Belgium*

^b*Verhaert – Masters in Innovation®Headquarters
Hogenakkerhoekstraat 21, B-9150 Kruibeke, Belgium*

Abstract

For elderly people fall incidents are life-changing events that lead to degradation or even loss of autonomy. Current fall detection systems are not integrated and often associated with undetected falls and/or false alarms.

In this paper, a social- and context-aware multi-sensor platform is presented, which integrates information gathered by a plethora of fall detection systems and sensors at the home of the elderly, by using a cloud-based solution, making use of an ontology. Within the ontology, both static and dynamic information is captured to model the situation of a specific patient and his/her (in)formal caregivers. This integrated contextual information allows to automatically and continuously assess the fall risk of the elderly, to more accurately detect falls and identify false alarms and to automatically notify the appropriate caregiver, e.g., based on location or their current task.

The main advantage of the proposed platform is that multiple fall detection systems and sensors can be integrated, as they can be easily plugged in, this can be done based on the specific needs of the patient. The combination of several

*Corresponding author; Tel: +32 9 331 49 38; Fax: +32 9 331 48 99

Email addresses: femke.debackere@intec.ugent.be (F. De Backere),
femke.ongenae@intec.ugent.be (F. Ongenae), floris.vandenabeele@intec.ugent.be
(F. Van den Abeele), jelle.nelis@intec.ugent.be (J. Nelis),
pieter.bonte@intec.ugent.be (P. Bonte), eli.clement@verhaert.com (E. Clement),
Matthew.Philpott@verhaert.com (M. Philpott), jeroen.hoebeke@intec.ugent.be
(J. Hoebeke), stijn.verstichel@intec.ugent.be (S. Verstichel),
ann.ackaert@intec.ugent.be (A. Ackaert), filip.deturck@intec.ugent.be (F. De Turck)

systems and sensors leads to a more reliable system, with better accuracy. The proof of concept was tested with the use of the visualizer, which enables a better way to analyze the data flow within the back-end and with the use of the portable testbed, which is equipped with several different sensors.

Key words: Fall Detection, Fall Risk Assessment, Ontology, Semantic, Context-aware

1. Introduction

For elderly people fall incidents are often life-changing events that might lead to degradation or even loss of autonomy. More than half of the elderly living in a nursing home and about one third of the elderly living at home fall at least once a year [1]. Of those who fall, 10 to 15% suffer severe injuries. The lack of timely aid can lead to further complications [2]. Although not all fall incidents lead to physical injuries, psychological consequences are equally important. Due to the high impact of falling, both fall prevention and reliable fall detection are necessities. The fall risk and detection work processes, as currently being employed, are visualized in Figure 1(a).

The increased risk factors that are related to falls can be divided into extrinsic and intrinsic risks¹. The first are directly related to the environment, while the second include, next to age and general clinical condition, sedentary behavior, impaired mobility and gait due to reduced muscle strength. To assess the risk of falling that a person has, formal caregivers carry out standardized tests. For example, the “Timed Get Up and Go Test (TUG)” is a test that is carried out in clinical environments on predefined, long time intervals [3]. Based on the results, targeted measures and advice are formulated. Further follow-up is done through direct interaction between a caregiver and the elderly. Today, attempts to automate this assessment and follow-up through domotics and monitoring systems are limited and not integrated [2].

When the help desk is notified of a fall, either by one of the fall detection sensors or because the elderly pushes the button, the help desk operator tries

¹http://www.euro.who.int/__data/assets/pdf_file/0018/74700/E82552.pdf

to contact the elderly to obtain extra information. When the elderly does not respond or help is needed, the appropriate (in)formal caregivers are contacted by phone, in a pre-defined order until someone is found who is able to tend to the elderly. The help desk thus does not take into account the current context, e.g., location or current tasks of the (in)formal caregivers to assign the alarm. This leads to unnecessary delays and distractions, as caregivers are often called although they are unavailable. Moreover, a caregiver who is far away is often called, while a nearby (in)formal caregiver, e.g., a neighbor, could more quickly assess the situation or even tend to the alarm. In case the elderly is injured as a result of the fall, quick response and intervention is of the utmost importance.

To counter undetected falls, false alarms and long intervention times, the Fall-Risk project² aims to develop a social- and context-aware multi-sensor framework, called the OCarePlatform, which is able to automatically and accurately assess the fall risk of the elderly, detect falls and assign the alarms to (in)formal caregivers by taking their current context into account. The home of the elderly is equipped with a plethora of sensors and one or more fall detection systems. The OCarePlatform collects and integrates all the monitoring data, gathered by these devices. Unorganized, unprocessed raw data can be voluminous, but has no meaning on itself as it has no relationships or context. Information is data that has been given meaning by defining relational connections. For this, the platform uses an ontology [4], which is a semantic model that formally describes the concepts in a certain domain, their relationships and attributes. This way, an ontology encourages re-use and integration. Knowledge models and more specifically ontologies are already used in medicine and care [5, 6, 7]. By managing the data about the current context in an ontology, intelligent algorithms can be more easily defined that take advantage of this information to accurately assess the fall risk and detect calls. The envisioned fall risk and detection work processes supported by the OCarePlatform are visualized in Figure 1(b).

First, by monitoring the behavior of the elderly, based on the integrated sensor information, the OCarePlatform automatically checks whether the elderly is compliant with the given advice. Moreover, the Activities of Daily Living (ADL)

²<http://www.iminds.be/en/research/overview-projects/p/detail/fallrisk-2>

patterns can be monitored. If a change in these patterns is detected, this might indicate a potentially higher risk for falls. In case of non-compliance or increased fall risks are detected, the appropriate caregivers are notified and informed of the gathered contextual information. This supports pro-active follow-up of the fall risk. The intervention of experts can thus be optimized in view of the time and point of need and the early warnings can contribute to increased effectiveness in view of fall prevention.

Second, the OCarePlatform also aims to reduce the amount of false alarms and undetected calls, by integrating the information, gathered by the fall detection systems with the available context information.

Third, the OCarePlatform also gathers context information about (in)formal caregivers associated with the elderly. It uses this information to more effectively assign one or more (in)formal caregivers to the alarm.

For example, when an alarm button is pushed, but the context information suggests that it is a false alarm with a high probability, the help desk operator can contact a close by neighbor or other person. After a specific time, the help desk operator will contact this informal caregiver to check if it was actually a false alarm and to exchange information about the gravity of the alarm. When a caregiver and help desk receive an alarm, they also receive all the context information, which is perceived as important at that time, e.g., the last known location of the elderly, his or her fall risk and the sensor or detection system that caused the alarm. Finally, feedback given by the caregiver network after an alarm situation or follow-up of fall risk assessment, initiated by a notification of the OCarePlatform, will allow the OCarePlatform to adapt to the needs and preferences of the caregiver and the elderly.

In this paper, a platform is presented capable of interacting with a wide range of heterogeneous sensors. Moreover, the platform is designed in such a way that intelligent, semantic techniques are used in order to deliver just-in-time information, about fall risk and fall detection, to the elderly and their (in)formal caregivers. The remainder of this paper is organized as follows. In Section 2, an overview is given of the current state of the art. Section 3 details the general architecture of the proposed FallRisk system. Illustrative scenarios are detailed

in Section 4 to highlight the possibilities of the proposed system and ontology-based reasoning. Sections 5, 6 and 7 further detail the different components of the system, namely the used continuous care ontology, the sensor-based monitoring system and the ontology-based OCarePlatform. Section 8 details the implemented Proof of Concept to demonstrate the feasibility of the proposed FallRisk system. Finally, Section 9 highlights the conclusions and future work.

2. Related Work

Systems and solutions developed to detect falls, can be divided into three different approaches based on the used data and sensors [8]. The first approach is based on the use of wearable devices and sensors, such as accelerometers and posture sensors. The second one is classified as ambient device based. In this approach, an attempt is made to create a fall detection system, by using video data, audio data and/or other event sensing. Finally, the last approach uses cameras to detect falls or analyze ADLs.

2.1. Wearable devices and sensors

This approach uses on-body sensors, for example, the Personal Alarm Systems (PAS). When an elderly is in need or falls, he or she pushes the button and a help desk is notified. However, the button is often accidentally pushed and the PAS is often not worn [2]. Some systems also incorporate built-in accelerometers to automatically detect movements and falls. However, the algorithms processing this accelerometer data do not detect all falls (false negatives) and/or generate false alarms (false positives).

Sensors can also be used in combination with each other, e.g., Wang [9] uses information from an accelerometer, cardiometer and smart sensors to analyze falls. Combining sensors can improve the accuracy. Tri-axial accelerometers [10] make it possible to recognize activities [11], thus improving fall detection systems. Pyroelectric InfraRed (PIR) sensors can also be used to classify different human activities [12]. Several techniques can be used to analyze the data, such as machine learning algorithms. Other solutions make use of the built-in accelerometers and gyroscopes of smartphones [13, 14]. Nowadays, clothing can

also be integrated with sensors, for example, T-shirt incorporated with ECG electrodes [15] or respiratory frequency detection [16]. GPS measurements could also provide added value in the recognition of activities and spatial context [17].

2.2. Ambient device based

Audio sensors, placed in the house, analyzing sound events can also be used [18, 19, 20, 21, 22]. For example, Litvak, et al. [23] use a microphone together with a floor vibration sensor and Li, et al. [24] use a circular array of microphones. However, only limited results on automated fall detection are reported. Another example is the use of floor-mounted accelerometers to gather signals that could identify a fall [25]. Sensors, from other devices, for example the Kinect’s infrared sensor can also be used to create a fall detection system [26]. Using integrated sensors of systems, already used by the patient, increases the uptake.

2.3. Vision based

Video systems can be used that observe the movements and behavior of the elderly. The main difference in the solutions within this approach, lie in the techniques to analyze the video streams together with the focus of the analysis, for example, posture analysis [27] or 3D head analysis [28]. Examples of different techniques are boosted key-frame selection together with correlated pyramidal motion-feature representation [29], spatio-temporal Laplacian pyramid coding [30], using motion and structure features [31] and shape analysis [32].

2.4. Our approach

Within our platform, different approaches can be integrated, based on the profile (e.g., cost, mobility, preferences) of the elderly. Moreover, it is possible to combine different approaches. The OCarePlatform, as described in Section 3, receives the result of one specific solution as input, independent from other detection systems that may be deployed within one home setting. Different techniques can send their analysis of a fall to the platform, which makes it possible to combine the results. This will improve flexibility, consistency and sensitivity. Furthermore, by using the platform information about the caregivers

and other context information, that becomes readily available, can be used to outline the situation of the elderly.

One could suggest that one of the biggest concerns to use these platforms is the intrusion of the elderly’s privacy. However, our experience within the focus groups, held in context of this research, which is confirmed by Berridge [33], is that elderly may use these technologies if it reduces caregiver burden and if it avoids having to move to assisted care facilities.

3. General Architecture of the FallRisk system

The general, high-level, architecture of the FallRisk project is visualized in Figure 2. As you can see on this figure, different acronyms are used on the arrows to indicate which type of data is processed by the FallRisk system and is sent to the Cloud and more specifically, the *OCarePlatform*.

Information about the residence of the elderly, e.g., sensor information, video camera images and micro arrays sound, are gathered on the *Local Gateway*. Thus, the processing of data from video and audio streams is done within the home. On the one hand, this secures the integrity of the data, as only results of an analysis are sent to the platform, on the other hand, this makes sure that the platform does not receive an overload of information. Also, information from the elderly’s smartphone, e.g., accelerometer information and GPS data. Some information will already be preprocessed in the home of the elderly, for example the processing of the video images, as sending all raw video data to the server is too bandwidth consuming. Also, the *Local Gateway* will be able to do some rigorous analysis of the data, in order to provide a back-up plan when the transmission of data to the servers is disabled due to for example malfunctioning of the internet connection. Data gathered by the *Local Gateway* is called raw data, or Care Data (CD). In this *Local Gateway*, CD is transformed into Meta Care Data (MCD), data enriched with information from the elderly and linked with other specific information, such as timestamps and identifiers.

Next, the *Local Gateway* will send MCD, through the internet to the Cloud and more specifically in the first place to the *Controllers*. Also other MCD, originating from outside the patient’s residence, can be transmitted to the *Controllers*.

This is a server-side system, managing the different connections between the various clients and the *OCarePlatform*. The *Controllers* are responsible for the following tasks:

- Collecting contextual information (MCD), originating from the smartphones of the (in)formal caregivers.
- Providing mechanisms for back-up of the transmitted MCD and storing it within the *Datasources*. The *Datasources* will also store all static information of the elderly patients and their (in)formal caregiver(s), their profiles, contact information and such.
- Adding one or multiple Meta Care Concepts (MCC) to MCD. MCC is in fact a tag to identify the corresponding concept from the ontology, used by the *OCarePlatform*. MCD, enriched with one or more MCC is called Meta Care Information (MCI). An example of this tagging is detailed in Section 6.
- Notifying the emergency response centre when a fall is detected, this is done based on the information provided by the *OCarePlatform*. The desk operator will then be responsible for notifying the correct (in)formal caregiver(s).

MCI, generated by the *Controllers* is sent to the *OCarePlatform*, which will infer new knowledge out of this MCI, by using ontology-based reasoning. This new knowledge is called Meta Care Knowledge (MCK). MCI and MCK, concerning contextual information and fall estimation and detection, is sent back to the *Controllers*. The *Controllers* are then responsible for notifying the correct (in)formal caregiver(s) or the emergency response centre. A more in-depth overview of the *OCarePlatform* is given in Section 7.

4. Scenario description

In order to show the possibilities of the proposed architecture and ontology-based reasoning methodology, this Section describes two illustrative use cases, i.e., (i) assessing and notifying a potential fall risk and (ii) detecting a fall and

notifying the appropriate (in)formal caregivers. How these scenarios are realized is discussed in detail in the following sections.

The scenario concerns an elderly woman Marie, who still lives at home. She has a son Tom, who lives nearby and checks in on her regularly. Tine, a formal caregiver, also visits Marie each morning to help her with her daily tasks. Marie also owns a Personal Alarm System (PAS), which she wears as a bracelet around her wrist. Consequently, accidental calls frequently occur.

4.1. Assessing and notifying a potential fall risk

During the last check-up with her care coordinator, it was assessed that Marie has great difficulty climbing stairs. This significantly increases her fall risk. Therefore, the care coordinator, Marie and Tom decide to install a chair lift. However, Marie finds the chair lift quite slow and tedious, so she often does not use it. Her (in)formal caregivers are unaware of this. One night, Marie gets up because she needs to go to the bathroom downstairs and she decides to descend the stairs herself. The motion detectors detect that Marie is moving in the area of the stairs. The *Local Gateway* gathers the information from the motion sensors, the pressure sensor in the chair lift and the light sensors in the hallway, which is dimly lit. It transforms this CD into MCD. The *Local Gateway* then sends this MCD to the *Controllers*. These transform the MCD to MCI by tagging it with an MCC from the ontology, in such a way that the *OCarePlatform* understands the meaning of the data. For example, the motion sensor data is tagged with the MCC `MotionObservation`. The *OCarePlatform* then processes this MCI. First, it decides that Marie is walking on the stairs without using the chair lift. Next, the *OCarePlatform* checks if this type of activity, i.e., walking on the stairs, entails a fall risk for this specific elderly. This is the case for Marie. Furthermore, the *OCarePlatform* reasons that the risk is even greater, because the area is badly lit. Based on this, the *OCarePlatform* informs the *Controllers* that a notification of a severe fall risk should be sent to Tine and Tom. The gathered context information about the fall risk is included in the notification. The *Controllers* decide to mail the notification to not disturb Tine and Tom during their sleep as the notification is not urgent. The next morning, Tine visits Marie and warns her again about the risks of

taking the stairs.

4.2. Detecting a fall and notifying the appropriate (in)formal caregivers

One evening, Marie goes to the kitchen and starts boiling some water to prepare tea. She walks to the closet to grab a cup, but it falls out the cupboard. She quickly ducks to try to catch the cup. This quick movement is picked up by the accelerometer in the smartphone of Marie. The *Local Gateway* receives this CD and transforms it to MCD. Meanwhile, the *Local Gateway* is collecting information from the other sensors in the house. Marie is still walking around in the kitchen. The *Local Gateway* sends this MCD and the accelerometer MCD to the *Controllers*, which tag it with MCC. This MCI is forwarded to the *OCarePlatform*. It first reasons on the accelerometer data and concludes that a potential fall has occurred. Next, it combines this conclusion with the other sensor data. As Marie is still moving around in the kitchen and no other person is present in the home, the *OCarePlatform* decides that it was a false alarm. The *OCarePlatform* stores this conclusion, but no further action is taken.

Meanwhile, Marie grabbed a cup of tea. She walks back to the living room to watch some TV. However, on her way there she trips and falls down. Luckily, she is wearing her PAS and she quickly pushes the button. Again, the accelerometer in her smartphone has also picked up the quick downward movement and MCD is sent to the *Controllers* and the *OCarePlatform* in the same manner as described in the previous paragraph. The *OCarePlatform* again decides that a potential fall has occurred. However, this time the *Local Gateway* has also forwarded the alarm from the PAS to the *Controllers* and *OCarePlatform*. As two systems have now indicated that a potential fall occurred, the *OCarePlatform* reasons on the available profile and context information of the (in)formal caregivers to decide in which order these caregivers should be contacted to handle the alarm. As the *OCarePlatform* knows that Tom lives nearby and is mostly at home in the evening, it decides that Tom should be notified first and then Tine. The *OCarePlatform* sends the alarm, its context information and the order in which the caregivers should be notified to the *Controllers*. They forward the alarm to the emergency response centre.

The desk operator receives the alarm. She also immediately sees the context

information about the call. The desk operator immediately calls Marie, who picks up her smartphone. Marie answers that she fell while holding a hot cup of tea and she cannot get up. Eva explains she will contact help and looks into the system, which suggests to contact Tom. She calls him and explains the situation. Tom agrees to drive to Marie and hangs up. Eva calls Marie again to let her know Tom is on his way.

In the meantime, Marie is still lying on the floor. Tom has a key to Marie's house. He enters and tries to see if she is wounded and helps Marie to stand up again. Marie has a small burn wound from the tea, but does not need urgent medical attention. Marie's phone rings and Tom answers the call. It is Eva again to check if everything is alright. Tom confirms this. Eva wishes both of them a nice evening and closes the call. She enters into the system that the fall was correctly detected and handled by Tom.

Note that the second person on the list, namely Tine, did not need to be notified. However, from the moment Eva notified the first person, a timer started running. When a pre-defined deadline is crossed, Eva is notified that she should call the second person on the list, because the first person has not checked back in with the help desk.

5. The ambient-aware continuous care ontology

In previous research [34], we created the ambient-aware continuous care ontology (ACCIO), which models knowledge and context information needed to optimize continuous care processes, using information technology. The modular ontology is split up in 7 high-level, generic ontologies and 2 types of low-level domain-specific ontologies. The high-level ontologies model knowledge that is applicable across all continuous care domains, i.e., hospitals, residential care and homecare, and is of interest to a plethora of healthcare applications and services. The seven high-level ontologies are: the *Upper*, *Sensor*, *Context*, *Profile*, *Role & Competence*, *Medical* and *Task* continuous care ontologies. These ontologies also import other existing ontologies, e.g., to model time (SWRL-

TemporalOntology³) and processes (OWL-S⁴). Two types of low-level ontologies modeling knowledge particular to specific domain were created, namely the low-level *Cure* and *Care* ontology. The first is tuned towards the knowledge exchanged in hospitals, while the second focusses on residential care and home-care. These ontologies extend the concepts defined in the high-level ontology, e.g., roles and competences, with knowledge specific to their domain, e.g., the specific roles and competences within a hospital and how they map on each other. More information about these ontologies can be found in [34].

The ACCIO ontology was extended with concepts, relations, properties and axioms, which are needed to model the used contextual sensors and fall detection systems, detected falls, falls alarms, fall risk estimations and the relevant profile information of the (in)formal caregivers and elderly. The most important concepts of this extended ACCIO ontology are shown in Figure 3.

The *Upper* ontology describes general classes, relations and axioms. All the other high-level ontologies import this ontology and define all their concepts as subconcepts of **Entity**. This is not shown on Figure 3 to avoid overload.

The *Sensor & Context* ontologies are used to describe all the observations made by the systems in the house of the elderly. The **System** concept models a system and its components, e.g., **Sensors** and **SensorBoards**. Note that the ontology models both the fall risk systems, e.g., **Micro Arrays** and **PAS**, as well as the environment, activity and position sensors, e.g., **Light Sensor** and **Motion Sensor**. An **Observation** represents a data value monitored by a system. **Symptom** concepts model axioms and rules, which allow detecting specific phenomena in the observations. Using OWL2 DL mechanisms, axioms are provided that reclassify these symptom individuals as **Fault** and **Solution** concepts. These ontologies also contains all the information related to localization. A **Location** either can be a **Coordinate** or a **Zone**. Finally, the **Furniture** available in the house and which **Systems** are integrated into them, is also represented.

The *Profile* ontology models the profile information about (in)formal caregivers and elderly. Each **Person** is associated with a **Profile**, which consists of a basic

³<http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTemporalOntology>

⁴<http://www.w3.org/Submission/OWL-S/>

and a risk profile. The latter, for example, allows to model that an elderly has a **Fall Risk**. The **Action** with which this risk is associated can also be modelled, e.g., **Taking Stairs**. The current **Status** of a person is also included, e.g., **Busy** or **Available**. Finally, the **Trust Relationship** between two people can be represented, which can be taken into account to assign an alarm, notification or task to a person.

The *Role & Competence* ontology allows to associate people with their **Roles**. The ontology defines each **Role** by its **Competences** through axioms. This supports algorithms that find the most appropriate staff members to fulfill a task based on the required competences.

Finally, the *Task* ontology models continuous care process workflows. A workflow represents a sequence of related continuous care tasks, which are conducted in a particular order. The **Process** concept models a process, which can return information and produce a change in environment based on the context and the information it is given. The **Process** concepts is further divided into **Actions** and **Tasks**. It is also modelled who performs these actions and who is assigned to task. Finally, the **Notification** concept is used to model notifications, which are generated based on the available context information, e.g., a situation with a fall risk has occurred or a fall has been detected.

6. Sensor-based monitoring system

One of the strengths of the presented *OCarePlatform* is its capability to process monitoring data coming from a variety of sensors. Such heterogeneous data enables more accurate assessment of fall risk and fall detection. In this section, first the different types of sensors and sensor systems that can provide relevant information to the *OCarePlatform* are presented. Next, another challenge that needs to be overcome in order to make the sensor data ready for further processing, namely the transformation of the raw data (CD) into MCD and MCI, is discussed.

6.1. Sensors

In order to provide contextual information, several sensors are installed within the home of the elderly. These sensors are listed in Table 1 and are grouped

according to the type of data they measure.

A first type of context information relates to the environment and includes parameters such as ambient temperature, humidity and light. As an illustrative example, the *OCarePlatform* could take into account that a badly lit room increases the risk of a fall and uses this information when processing any subsequent fall events. For the second type of context information, passive infrared motion sensors collect data about the activity level of the elderly. Long-term changes in the activity level can be taken into account by the *OCarePlatform* when assessing the fall risk. Furthermore, when a fall is detected by a fall detection system the *OCarePlatform* can use the motion context, either to filter out false positives or to locate the place where the fall occurred.

Pressure sensors, located in e.g. a bed, a chair or a couch, provide a last type of context information, namely exact position information. They help the *OCarePlatform* to determine the exact location of the elderly, when he or she is sitting or lying down (in this situation motion sensors have a hard time detecting the elderly). When a fall detection system reports a fall and the data from the pressures sensors reveals that just before the fall, the person was sitting on a chair, the *OCarePlatform* is able to identify the cause of the fall.

The upper-left part of Figure 4 illustrates how the sensors from Table 1 could be placed throughout a service flat, although they could also be retrofitted into an elderly's home. Motion sensors in the living and bathroom track the elderly's activity while pressure sensors check the occupancy of the bed, the couch and the toilet. Environment sensors track the ambient living conditions in the living room and possibly in the bathroom as well (not shown in figure). Also note the presence of the *Local Gateway*. All data generated by the sensors passes through this *Local Gateway* before it can be transmitted to the *OCarePlatform*.

Apart from sensors collecting context information, dedicated fall detection systems also provide important data to the platform. Three separate systems for detecting falls are detailed here. Two of these systems are also used for fall risk assessment.

The first system detects falls by use of video camera's. In this case, video images shot by camera's throughout the house are fed to the *Local Gateway* for

processing. A combination of person tracking and detecting multiple features relating to the posture of the person is used to determine whether a fall has occurred [35]. Tracking a person in video images also allows to continuously estimate the fall risk: e.g. when video images indicate that a person’s activity level has decreased, the *OCarePlatform* could decide to increase the fall risk and act accordingly.

The second option is that of a wearable Personal Alarm System (PAS), which was already explained in Section 1. The third fall detection system employs a set of microphones that are positioned in the corners of a room. In case of a fall, the array of microphones is able to recognize the pattern of a fall in the mechanical waves that it receives [24]. Microphones arrays are also suitable for monitoring the walking habits of a patient. Changes in walking habits (like shuffling their feet) are useful to assess the fall risk.

It is clear from the above discussion that a multitude of sensors and sensor systems exist that can help to improve fall detection and fall risk assessment. Of course, not every home will be equipped with all sensors described here. In reality available sensor combinations will vary per installation or may evolve over time. As such, it is important to have a platform that is capable of dealing with this heterogeneity and getting the most out of the available information.

6.2. Transforming CD from sensors into MCD and MCI

The raw data generated by the sensors, called CD, is not suitable for further processing by the *OCarePlatform*, which requires MCI as detailed in Section 3. For instance, the detection of motion in itself is useless as long as it cannot be bound to a particular location, home and time. To overcome this, a number of translation steps are required to transform CD into MCI. Briefly summarized, this process involves the *Local Gateway* in the home, which is responsible for gathering CD from the different sensors, transforming it to MCD and passing the result to the *Controllers*. The *Controllers* will then tag the MCD with a MCC to generate MCI. Later in this Section, this process is illustrated for CD generated by a motion sensor. Although this process may look simple, its realization is far from trivial mainly due to the wide variety of sensor solutions that exists, many of these solutions, using their own proprietary protocols and

data formats. In this Section, the transformation from CD to MCD and MCI, when using state-of-the-art embedded Web Service technology and open data formats, is discussed.

As Web Service technology was at the basis of the great success of the current Internet, the Internet Engineering Task Force (IETF) has decided to work on the standardization of a framework for the realization of RESTful embedded Web Services in a similar way as traditional Web Services, but suitable for the most constrained nodes (i.e. sensors) and networks. Their work resulted in the Constrained Application Protocol (CoAP), a specialized RESTful web transfer protocol for use with constrained networks and nodes. Together with 6LoWPAN and UDP, CoAP allows the integration of sensors into the Internet and in Web Services in a standardized way [36]. CoAP uses the same RESTful principles as HTTP, but it is much lighter so that it can be run on constrained devices. As a result, CoAP has a much lower header overhead and parsing complexity than HTTP. Using CoAP, a sensor can be turned into an embedded web server, offering access to its resources through URIs. For instance, a motion sensor may offer a resource with URI path `"/sen/0/status"`. When a client issues a CoAP GET request for this resource, the sensor will return the current status of the motion sensor.

Next to this, CoAP has a number of very interesting extensions, which will help the system to collect and convert sensor data. First of all, CoAP offers an observe mechanism, allowing a client to be notified about every change in resource state. For instance, when observing a motion sensor, a client will be automatically notified about every motion event. This way, it becomes very straightforward to collect sensor data e.g. at the *Local Gateway*. Secondly, every CoAP device also has a well-known URI `"/.well-known/core"`. Through this URI, a list of available resources can be obtained together with their attributes. These attributes reveal additional information about the resource, such as the resource type and the interface (i.e. which methods are supported: GET, PUT, etc.). For example, Listing 1 shows the result of querying `"/.well-known/core"` of a motion sensor that has, next to a resource for the motion sensor status, a resource with the location where the sensor has been installed. The values of the

attributes is based on the IPSO Application Framework⁵. Finally, there is also the concept of a Resource Directory (RD), which hosts descriptions of resources held on other CoAP servers that registered with the RD, allowing lookups to be performed for those resources. For instance, a RD on the *Local Gateway* will automatically retrieve an overview of all available CoAP sensors and their resources.

Listing 1: Retrieving all resources on a CoAP sensor

```
</sen/0/status>;rt="ipso.sen.mot.status;if="core#s",  
</loc/sem>;rt="ipso.loc.sem";if="core#s"
```

The above mechanisms provide all ingredients needed to easily convert the data generated by the sensors into MCD and MCI. The *Local Gateway* implements a RD. All sensor nodes will automatically register themselves with this RD. Now the gateway has an overview of all available sensors. Next to this, the gateway can easily find out the type of sensor, using the resource attributes, and additional information, such as the location, provided by other resources. From that point on, the gateway can automatically start collecting all sensor data using e.g. the observe mechanism described above.

Now, when Marie walks down the stairs, the motion sensor in the hallway senses a change in motion and reports this change by transmitting the following message, containing CD to the gateway, as shown in Listing 2. This message is formatted in JSON, according to the Sensor Markup Language⁶ (SenML+JSON). A value of “1” indicates that motion has been detected and a “0” indicates that the sensor stopped detecting motion. Again, the modeling of the information reported by the different types of sensors is based on the IPSO Application Framework⁷. Using standards to model this information facilitates further processing.

Listing 2: Example of Care Data for a motion sensor

```
{  
  "e": [{ "n": "sen/0/status", "v": 1, "rt": "ipso.sen.mot.status" } ]  
}
```

⁵<http://www.ipso-alliance.org/downloads/Application+Framework>

⁶<http://tools.ietf.org/html/draft-jennings-senml-10>

⁷<http://www.ipso-alliance.org/downloads/Application+Framework>

This information arrives at the gateway, which now generates MCD from CD by adding a timestamp, a patient identifier, a unique data source identifier (hardware ID of the sensor) and data source type. In addition, using the information provided by the "/loc/sem" resource on the motion sensor, the gateway is able to automatically add location info to the data. Note that if no location information is available on the sensor itself, the gateway needs to be configured with the locations of the sensors throughout the house. This results in the following MCD, shown in listing 3.

Listing 3: Meta Care Data example for a motion sensor

```

{
  "timeStamp": "2013-09-20T11:19:27+02:00",
  "userID": "CFFF2C57-D52E-4782-A785-79A4B66C1841",
  "dataSourceID": "AA:BB:CC:DD:EE:FF",
  "dataSourceType": "http://occs.televic.com/data-sources/SensorBoard",
  "data": [
    {
      "e": [
        { "n": "sen/0/status", "v": 1, "rt": "ipso.sen.mot.status"},
        { "n": "loc/sem", "v": "hallway", "rt": "ipso.loc.sem"}],
      "bn": "coap://[aaaa::3]:5683/"
    }
  ]
}

```

Next, the MCD is sent to the *Controllers* which will tag it with one or more MCCs from the ontology, e.g., *MotionSensor*. Listing 4 show the resulting MCI.

Listing 4: Meta Care Information example for a motion sensor

```

{
  "prefixes" : {
    "profile" :
      "http://server.be/ontologies/AccioOntology/ProfileAccio.owl#",
    "wsna" :
      "http://server.be/ontologies/AccioOntology/WSNAdjusted.owl#",
    "ca" :
      "http://server.be/ontologies/AccioOntology/ContextAccio.owl#",
  }
  "timeStamp": "2013-09-20T11:19:27+02:00",
  "userID": "profile:CFFF2C57-D52E-4782-A785-79A4B66C1841",
  "dataSourceID": "wsna:AABBCCDDEEFF",
  "dataSourceType": "wsna:SensorBoard",
  "data": [
    {
      "e": [
        { "n": "sen/0/status", "v": 1, "rt": "ipso.sen.mot.status",
          "ontology-uri": "wsna:AABBCCDDEEFF.sen.mot.status",
          "tag": "wsna:MotionSensor"},
      ]
    }
  ]
}

```

```

    { "n": "loc/sem", "v": "hallway", "rt":
      "ipso.loc.sem", "ontology-uri":
        "wsna:AABBCCDDEEFF.loc.sem", "tag": "ca:Location"}],
    "bn": "coap://[aaaa::3]:5683/"
  }
}

```

These translation steps are necessary to go from sensor data to information that the *OCarePlatform* can use for reasoning. Again, by using open standards the *Controllers* (which can be owned by a different actor than the sensors) is able to parse and process the data coming from the *Local Gateway*. Note that for brevity MCI employs a concept similar to XML namespaces: JSON prefixes.

It is clear that using open standards greatly facilitates the process of turning raw sensor data into MCD and MCI. In reality, not all sensor systems will adopt such open standards. As we strive to be able to incorporate as much heterogeneous sensor data as possible into the *OCarePlatform*, a solution has been designed to be able to incorporate other sensors as well. The *Local Gateway* is responsible to support sensors that do not natively support the format used by the FallRisk system. In previous research [37], a generic interoperability and integration platform, named DYAMAND was presented. DYAMAND enables devices using different technologies to be used by applications in a technology-agnostic way. The only information the application needs is which device type it is interested in, e.g. a motion sensor is presented to the application with a specific interface irrespective of the actual communication technology used. Using this DYAMAND platform, commercially available sensors can easily be integrated in the FallRisk system. The *Local Gateway* running an instance of DYAMAND will translate all events coming from a sensor in the house into an MCD that can be processed by the *Controllers*.

7. The *OCarePlatform*: Context-aware fall risk and detection notification system

Potentially, large amounts of data are sent to the *OCarePlatform*, mostly by the *Controllers*. However, any data producing entity could potentially interface with the *OCarePlatform*. The data thus originates from various sources inside and outside the different home settings. The *OCarePlatform* facilitates the

intelligent and coordinated integration, analysis, combination and efficient usage of all this MCI, by using the ontologies described in Section 5. A closer view on the *OCarePlatform* is shown in Figure 4 and is presented in more detail in this Section.

MCI enters the *OCarePlatform* through the *Gateway Service*, as depicted at the bottom of Figure 4. This service passes the MCI to the *Matching Service*. To be able to integrate the MCI, analyze it and draw conclusions from it, the MCI needs to be transformed into context information, i.e., ontology A-Box individuals from the continuous care ontologies used by the *OCarePlatform*. To perform this conversion, the *Matching Service* takes advantage of the tags in the communicated MCI. As the tags, i.e., MCCs, map on the concepts in the ontology, the *Matching Service* knows which type of individuals need to be created and forwards the MCI to the appropriate *Context Provider Service*. The *Context Provider Services* know how particular types of individuals should be created by analyzing the axioms defined in the ontologies. Different types of *Context Provider Services* are created for different types of individuals, e.g., sensor information or a detection of a fall. Next, this ontological data is published onto the *Semantic Communication Bus (SCB)*.

A large amount of generated data needs to be processed by the *OCarePlatform* in an efficient and manageable manner. Therefore, the *OCarePlatform* is developed as a modular platform, which consists of an extensible set of services, which process the incoming data. These services are referred to as *MCI Services*. They each use (a subset of) the ontologies described in Section 5 to perform their reasoning and realize their goal. To implement the scenarios described in Section 4, the following *MCI Services* were designed. The *Accelerometer Monitoring MCI Service* is responsible for processing the accelerometer data from the smartphone of the elderly and the *Stairs Monitoring MCI Service* monitors whether someone is moving around on the stairs and how. The *Fall Detection MCI Service* is responsible for reasoning on the data to decide whether a fall has occurred, while the *Fall Risk MCI Service* is responsible for assessing the risk a particular elderly has for falling. The *Help Selection MCI Service* finds the most appropriate caregiver to assign to a task or an alarm, based on the currently available context information. Finally, the *Notification MCI Service*

is responsible for sending the appropriate notifications, i.e., the results of the reasoning of the other *MCI Services*, to the *Controllers*.

Consequently, there is a need for an intelligent filtering system, capable of sending only that specific data to the *MCI Services*, in which they are interested at that time, keeping the overhead for the other services as low as possible. Moreover, this filtering system needs to operate as a communication substrate, glueing the different *MCI Services* together and orchestrating collaboration. For this, the *SCB* [38] was designed, which orchestrates the communication of semantically enriched data, by using the high-level continuous care ontologies, described in Section 5, as *Core Ontologies*. These ontologies make it possible to filter the data, based on semantics of the data, instead of on syntactical text patterns. The *MCI Services* use a *Context Manager*, which contains (a subset of) the core ontologies used by the *SCB*, to specify the context they are interested in, by defining filtering rules and registering them with the *Context Disseminator*. For example, the *Stairs Monitoring MCI Service*, which is interested in information produced by the chairlift as well as movement information in the area of the staircase, registers the following filter rule:

Event **and** hasContext **some** ((PressureStatusObservation **and** (isPartOf **some** ChairLift)) **or** (MotionObservation **and** (hasLocation **some** Staircase)))

Another example is the *Fall Risk MCI Service*, which registers following rule to express its interest in the actions of the resident and the light intensity in the rooms:

Event **and** hasContext **some** ((Action **and** (isPerformedBy **some** (hasRole **some** Patient))) LichtIntensityObservation)

The advantage of using an ontology-based semantic approach immediately becomes clear from the above examples. Although the *Fall Risk MCI Service* has not explicitly specified that it is interested in chairlift information, the core ontologies classify the usage of the chairlift as an **TakingStairs Action** of the resident. The data produced by the chairlift is therefore offered to the *Fall Risk MCI Service*, as well as to the *Stairs Monitoring MCI Service*.

The ultimate baseline rationale for adopting a bus-based architecture is the flexibility it brings to connect new innovative information processing services. Additionally, by supporting a semantic annotation mechanism, the connected services can specify in great detail the semantics of the data they are interested in and not only the syntax. This allows to reduce the amount of data that is forwarded to the *MCI Services*, which prevents them from being flooded with huge amounts of data. It also facilitates an agile approach, where new services can be more easily deployed or duplicated for scalability and redundancy.

An additional advantage of using a semantic-aware approach for the *SCB* is that the information distributed by this bus readily conforms to the ontologies adopted by the *MCI Services*. These *MCI Services* are the brains of the platform, the processing engines. Each of these services has a specific task. These tasks can be implemented in a number of ways. Firstly and most naturally, description logics classification axioms can be specified in the ontology to classify the incoming information and link it to appropriate action to be taken. Secondly, rule engines can be adopted to perform more complex, e.g., mathematical analysis, of the information. Lastly, proprietary algorithms can still be implemented as well. The ontology still ensures that the correct semantics are maintained. It is important to note that all conclusions, called MCK, drawn by the *MCI Services* are put back on the *SCB*. In this way, conclusions drawn by one *MCI Service* can be used by a second *MCI Service* as additional situational information. As such, the *OCarePlatform* supports the composition of complex services from a set of smaller services in a loosely coupled manner. The simple services perform specific reasoning tasks in parallel and notify their conclusions to other services, which have expressed an interest in this kind of information.

A specific example could be the *Stairs Monitoring MCI Service* concluding that the resident has taken the stairs, because it detected the movement of the elderly on the stairs while no pressure measurements on the chairlift have been detected. The result is a new MCK, representing the action **TakingStairs**, which is published back on the *SCB* by this *MCI Service* by using the *Context Manager*. Subsequently, this information is picked up by the filter rule of the *Fall Risk MCI Service*, which analyzes the environment at the time of taking the stairs. A potentially increased risk of falling could result when the resident takes

the stairs, but does not use the lights or takes a long time to actually complete the staircase. Again, this conclusion is fed back to the *SCB*, for example as a *FallRisk*, which is picked up by the *Help Selection MCI Service*. This *MCI Service* decided which people should be notified of this risky situation, based on their profile information and preferences, e.g., Tine and Tom, and feeds this knowledge to the *SCB* as for example a *FallRiskNotification*. Finally, this published data can then be taken up by, e.g., the *Notification MCI Service*, which is responsible for communicating the knowledge and conclusions, i.e., the MCK, deduced by the *OCarePlatform* with the *Controllers*.

8. Implementation of the Proof of Concept

In the previous sections, we elaborated in detail on the different components of the *FallRisk* system. In order to prove the viability of our proposed architecture and to evaluate the feasibility of injecting multi-sensor data into a reasoning system, the different components of the architecture have been implemented, integrated and tested.

To realize the sensor monitoring system, a portable embedded sensor network has been developed that emulates a smart home environment. This network has been integrated into a portable flight case, as shown in Figure 5. The setup consists of two parallel wooden panels that are spaced about 30 centimeters apart. The bottom panel incorporates the embedded devices and the gateway. The actual sensors and actuators are attached to the upper panel, where user interaction takes place. The gateway will observe all discovered sensors, translate the generated CD into MCD and forward this MCD to the *Controllers*. By interacting with the sensors and actuators on the upper panel, specific CD can be generated in a controlled way and can be used for evaluating the other components of the architecture, in particular the reasoning. This portable testbed realizes a controlled sensor monitoring system.

Next to this, a small apartment was created within our lab environment to evaluate the system in a controlled setting. For this, commercially available sensors were used, that were integrated with the *OCarePlatform*, using *DYAMAND*.

On the one hand, the Zenio fall detector, as shown in Figure 6 from Verhaert⁸ was used. This device can be worn as a necklace, which makes it very discrete and reliable. Alarms can be sent through cell phone. On the other hand, an Android smartphone application was developed, as visualized in Figure 7. This app is capable of linking with a Zenio fall detector, but also the built-in sensors are used to detect a fall even more accurately. When a fall is detected through the app, it is sent to the DYAMAND platform, which will forward the alarm, as MCD, to the *Controllers*. Next, the *Controllers* will transform MCD into MCI and forward it to the *OCarePlatform*, which is responsible for analyzing the data and deciding if it truly was a fall. This can be achieved by combining all the sensor and context information that is constantly gathering within the *OCarePlatform*.

As discussed earlier, the *OCarePlatform* consists of a large number of atomic services, each offering one, particular functionality. These atomic services can then be combined into complex workflows, such that the *OCarePlatform* can ideally support the caregivers and care receivers. Every care receiver has another subset of sensors installed is in the home tuned to his or her particular needs and fall risk. As such, each care receiver is subscribed to a different set of services. The service subscriptions and combination into workflows can also differ, based on the particular needs of this person and his support network. Consequently, the particular implementation of a service and combination into workflows is personalized for every elderly.

To derive which services are of importance to which types of residents and how these should be combined, a user-driven, interdisciplinary approach was adopted [34]. Workshops are organized in a realistic mock-up of a homecare setting and targeted end-users, e.g., elderly, (in)formal caregivers and representatives from healthcare organizations, are asked to role-play and discuss fall (risk) scenarios. These scenarios are supported by the *OCarePlatform*. When different scenarios are played with different resident profiles, different services and workflows are triggered, leading to different results and outcomes, e.g., a nurse might be notified in one scenario, while no notification is sent in another.

⁸<http://www.verhaert.com/>

This triggers feedback and discussion from the stakeholders whether this functionality is desirable. These workshops are supported by both social scientists and software developers.

To make it clear to the workshop participants, researchers and end-users why the *OCarePlatform* made particular decisions in certain situations, a visualizer tool was conceived that graphically illustrates the reasoning process. It visualizes which services are triggered by different scenarios, how are they combined, which data is exchanged between them and which processing is performed by each *MCI Service*. This visualizer tool, together with the portable sensor testbed, enables the interdisciplinary team to demonstrate how the reasoning process of the *OCarePlatform* differs between different scenarios and helps to foster discussion and feedback on particular *MCI Services* instead of on the *OCarePlatform* as a whole.

One of the main components of the visualizer is shown in Figure 8. The bars indicate how many MCI packets are residing within each component of the *OCarePlatform*. New components are automatically added to the user interface, when they are used in the *OCarePlatform*. This gives a high-level overview of the behaviour of MCI and MCK in the platform. It is possible to click on each bar and get a list of all the MCI and MCK, residing in one specific component. A specific *MCI Services* can then be selected to get a detailed overview of a specific MCI packet in a specific component. Figure 8 also shows the information of an MCI packet and visualises if reasoning is performed within the component. A detailed overview of the executed SPARQL queries is also shown. By looking at this visualization, workshop participants can thus get an idea of how data flows through the different components of the *OCarePlatform* and which reasoning, i.e., processing, is performed on the data in each component.

9. Conclusion

A social- and context-aware multi-sensor platform was presented, which integrates information gathered by a plethora of fall detection systems and sensors at the home of the elderly, by using an ontology. This integrated contextual information allows to automatically and continuously assess the fall risk of the

elderly, to more accurately detect falls and identify false alarms and to automatically notify the appropriate caregiver, e.g., based on location or current task.

It is important to note that presented platform does not need to be equipped with all the possible fall detection systems and available sensors to result in a reliable fall risk assessment and detection set-up. Depending on the needs and preferences of the elderly, the most appropriate devices should be selected that complement each other and lead to a reliable set-up for continuous monitoring. In order to prove the viability of our proposed architecture and to evaluate the feasibility of injecting multi-sensor data into a reasoning system, the different components of the architecture have been implemented, integrated and tested in a Proof of Concept. Future work will focus on how the platform and the profile and context information it gathers can be leveraged to give a suggestion of which systems and sensors should be integrated in the house of a particular elderly.

Acknowledgment

The iMinds FallRisk project is cofunded by iMinds (Interdisciplinary Institute for Technology), a research institute founded by the Flemish Government. Companies and organizations involved in the project are COMmeto, Televic Healthcare, TP Vision, Verhaert and Wit-Gele Kruis Limburg, with projectsupport of IWT.

References

- [1] K. Milisen, E. Detroch, K. Bellens, et. al, Falls among Community-Dwelling Elderly: A Pilot Study of Prevalence, Circumstances and Consequences in Flanders, *Tijdschr Gerontol Geriatr* 35 (1) (2004) 15–20.
- [2] F. Vlaskamp, J. van der Heijden, SeniorWatch: Fall Detector: A Cooperation between The United Kingdom and The Netherlands, Tech. rep. (December 2001).

- [3] D. Podsiadlo, S. Richardson, The Timed "Up & go": A Test of Basic Functional Mobility for Frail Elderly Persons, *J Am Geriatr Soc.* 39 (2) (1991) 142–148.
- [4] T. Gruber, A Translation Approach to Portable Ontology Specifications, *KNOWL ACQUIS.*
- [5] B. Pandey, R. Mishra, Knowledge and intelligent computing system in medicine, *Computers in biology and medicine* 39 (3) (2009) 215–230.
- [6] M. C. Díaz-Galiano, M. T. Martín-Valdivia, L. Ureña-López, Query expansion with a medical ontology to improve a multimodal information retrieval system, *Computers in Biology and Medicine* 39 (4) (2009) 396–403.
- [7] F. Ongenaes, M. Claeys, W. Kerckhove, T. Dupont, P. Verhoeve, F. De Turck, A self-learning nurse call system, *Computers in biology and medicine* 44 (2014) 110–123.
- [8] M. Mubashir, L. Shao, L. Seed, A survey on fall detection: Principles and approaches, *Neurocomputing* 100 (2013) 144–152.
- [9] J. Wang, Z. Zhang, B. Li, S. Lee, R. S. Sherratt, An enhanced fall detection system for elderly person monitoring using consumer home networks, *Consumer Electronics, IEEE Transactions on* 60 (1) (2014) 23–29.
- [10] P. Mostarac, R. Malaric, M. Jurcevic, H. Hegedus, A. Lay-Ekuakille, P. Vergallo, System for monitoring and fall detection of patients using mobile 3-axis accelerometers sensors, in: *Medical Measurements and Applications Proceedings (MeMeA)*, 2011 IEEE International Workshop on, IEEE, 2011, pp. 456–459.
- [11] L. Atallah, B. Lo, R. King, G.-Z. Yang, Sensor positioning for activity recognition using wearable accelerometers, *Biomedical Circuits and Systems, IEEE Transactions on* 5 (4) (2011) 320–329.
- [12] X. Luo, T. Liu, J. Liu, X. Guo, G. Wang, Design and implementation of a distributed fall detection system based on wireless sensor networks, *EURASIP Journal on Wireless Communications and Networking* 2012 (1) (2012) 1–13.

- [13] J. Dai, X. Bai, Z. Yang, Z. Shen, D. Xuan, Perfalld: A pervasive fall detection system using mobile phones, in: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on, IEEE, 2010, pp. 292–297.
- [14] B. H. Dobkin, A. Dorsch, The promise of mhealth daily activity monitoring and outcome assessments by wearable sensors, *Neurorehabilitation and Neural Repair* 25 (9) (2011) 788–798.
- [15] N. Noury, A. Dittmar, C. Corroy, R. Baghai, J. Weber, D. Blanc, F. Klefsat, A. Blinovska, S. Vaysse, B. Comet, Vtamm—a smart clothe for ambulatory remote monitoring of physiological parameters and activity., in: Conference proceedings:... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, Vol. 5, 2003, pp. 3266–3269.
- [16] M. Di Rienzo, F. Rizzo, G. Parati, G. Brambilla, M. Ferratini, P. Castiglioni, Magic system: A new textile-based wearable device for biological signal monitoring. applicability in daily life and clinical setting, in: Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, IEEE, 2005, pp. 7167–7169.
- [17] A. Subramanya, A. Raj, J. A. Bilmes, D. Fox, Recognizing activities and spatial context using wearable sensors, arXiv preprint arXiv:1206.6869.
- [18] D. Maunder, E. Ambikairajah, J. Epps, B. Celler, Dual-microphone sounds of daily life classification for telemonitoring in a noisy environment, in: Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, IEEE, 2008, pp. 4636–4639.
- [19] A. Temko, C. Nadeu, D. Macho, R. Malkin, C. Zieger, M. Omologo, Acoustic event detection and classification, in: *Computers in the Human Interaction Loop*, Springer, 2009, pp. 61–73.
- [20] A. Fleury, et. al, Improving Supervised Classification of Activities of Daily Living Using Prior Knowledge, *IJEHMC*.

- [21] H. Lozano, I. Hernandez, A. Picon, J. Camarena, E. Navas, Audio classification techniques in home environments for elderly/dependant people, in: *Computers Helping People with Special Needs*, Springer, 2010, pp. 320–323.
- [22] K.-Y. Huang, C.-C. Hsia, M.-s. Tsai, Y.-H. Chiu, G.-L. Yan, Activity recognition by detecting acoustic events for eldercare, in: *6th World Congress of Biomechanics (WCB 2010)*. August 1-6, 2010 Singapore, Springer, 2010, pp. 1522–1525.
- [23] D. Litvak, Y. Zigel, I. Gannot, Fall detection of elderly through floor vibrations and sound, in: *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, IEEE, 2008*, pp. 4632–4635.
- [24] Y. Li, , K. C. Ho, M. Popescu, A Microphone Array System for Automatic Fall Detection, *IEEE Transactions on Biomedical Engineering* 59 (5) (2012) 1291–1301.
- [25] F. Werner, J. Diermaier, S. Schmid, P. Panek, Fall detection with distributed floor-mounted accelerometers: An overview of the development and evaluation of a fall detection system within the project ehome, in: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on, IEEE, 2011*, pp. 354–361.
- [26] G. Mastorakis, D. Makris, Fall detection system using kinect’s infrared sensor, *Journal of Real-Time Image Processing* (2012) 1–12.
- [27] N. Thome, S. Miguet, A hmmm-based approach for robust fall detection., in: *ICARCV, Vol. 6, 2006*, pp. 1–8.
- [28] C. Rougier, J. Meunier, Demo: Fall detection using 3d head trajectory extracted from a single camera video sequence, *Journal of Telemedicine and Telecare* 11 (4).
- [29] L. Liu, L. Shao, P. Rockett, Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition, *Pattern Recognition* 46 (7) (2013) 1810–1818.

- [30] X. Zhen, L. Shao, D. Tao, X. Li, Spatio-temporal laplacian pyramid coding for action recognition, *IEEE Transactions on Cybernetics*.
- [31] X. Zhen, L. Shao, D. Tao, X. Li, Embedding motion and structure features for action recognition., *IEEE Trans. Circuits Syst. Video Techn.* 23 (7) (2013) 1182–1190.
- [32] L. Shao, L. Ji, Y. Liu, J. Zhang, Human action segmentation and recognition via motion and shape analysis, *Pattern Recognition Letters* 33 (4) (2012) 438–445.
- [33] C. Berridge, Seeing the social in technology for older adults: Making the implicit explicit through a multidisciplinary lens, in: *The Collective Spirit of Aging Across Cultures*, Springer, 2014, pp. 173–189.
- [34] F. Ongenaes, L. Bleumes, N. Sulmon, et. al, Participatory Design of a Continuous Care Ontology: Towards a User-Driven Ontology Engineering Methodology, in: *Proc. of International conference on Knowledge Engineering and Ontology Development*, 2011, pp. 81–90.
- [35] G. Debard, P. Karsmakers, M. Deschodt, et. al, Camera-based Fall Detection using Multiple Features validated with Real Life Video, in: *Proc. of International Conference on Intelligent Environments*, 2011, pp. 441–450.
- [36] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. V. d. Abeele, E. D. Poorter, I. Moerman, P. Demeester, Ietf standardization in the field of the internet of things (iot): A survey, *Journal of Sensor and Actuator Networks* 2 (2) (2013) 235–287.
- [37] J. Nelis, T. Verschueren, D. Verslype, C. Develder, Dyamand: Dynamic, adaptive management of networks and devices, in: *Local Computer Networks (LCN)*, 2012 IEEE 37th Conference on, 2012, pp. 192–195.
- [38] J. Famaey, S. Latré, J. Strassner, et. al, An Ontology-Driven Semantic Bus for Autonomic Communication Elements, in: *Proc. of the 5th IEEE international conference on Modelling autonomic communication environments*, 2010, pp. 37–50.

Tables and Figures

Data	Measured by
Environment	Ambient temperature, relative humidity & light intensity sensors
Activity	Passive infrared motion sensor
Position	Pressure sensor (contact plate)

Table 1: Monitoring sensors used for collecting contextual information

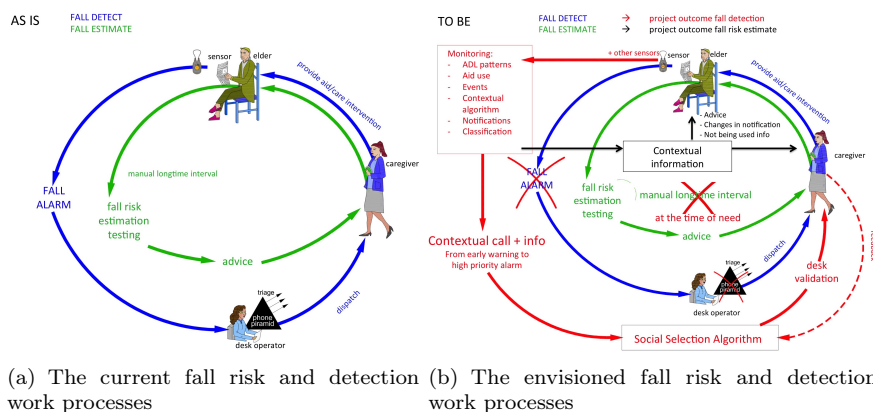


Figure 1: Comparison between the current and envision fall risk and detection work processes

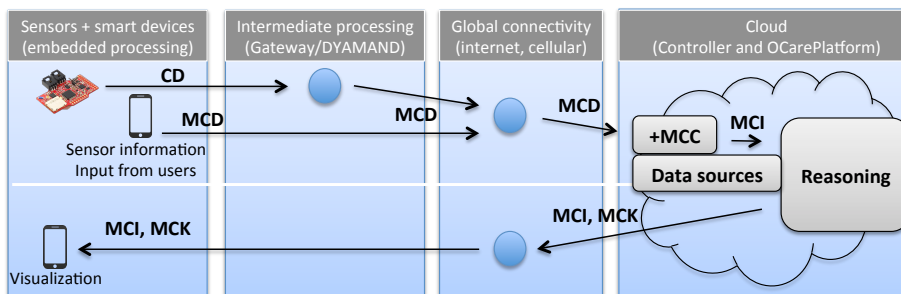


Figure 2: General, high-level, architecture of the FallRisk system

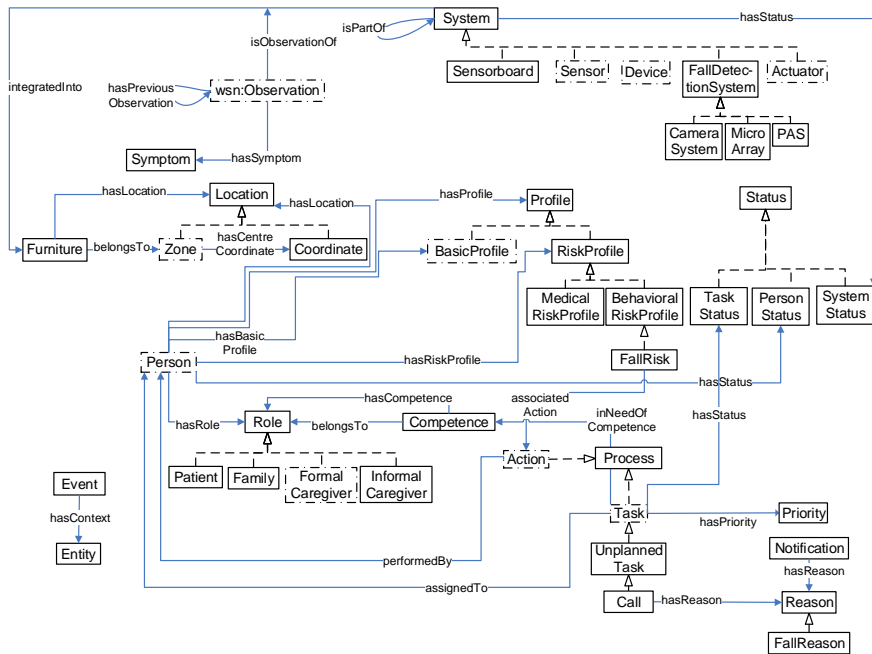


Figure 3: Prevalent concepts of the ambient-aware continuous care ontology for fall risk estimation and fall detection

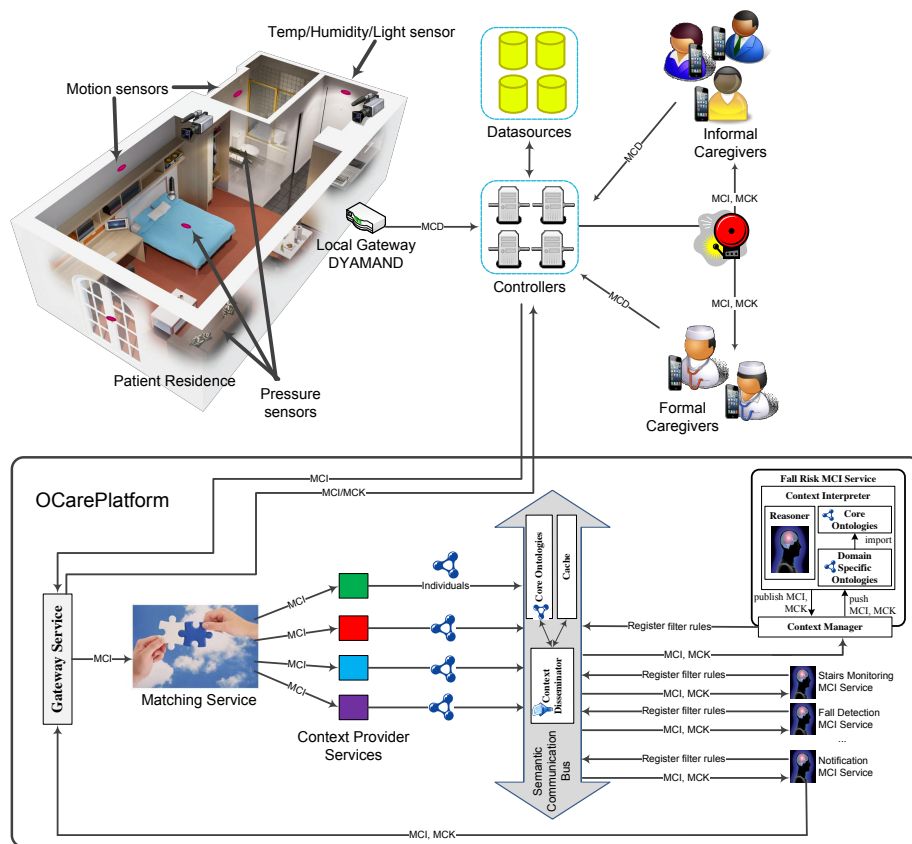


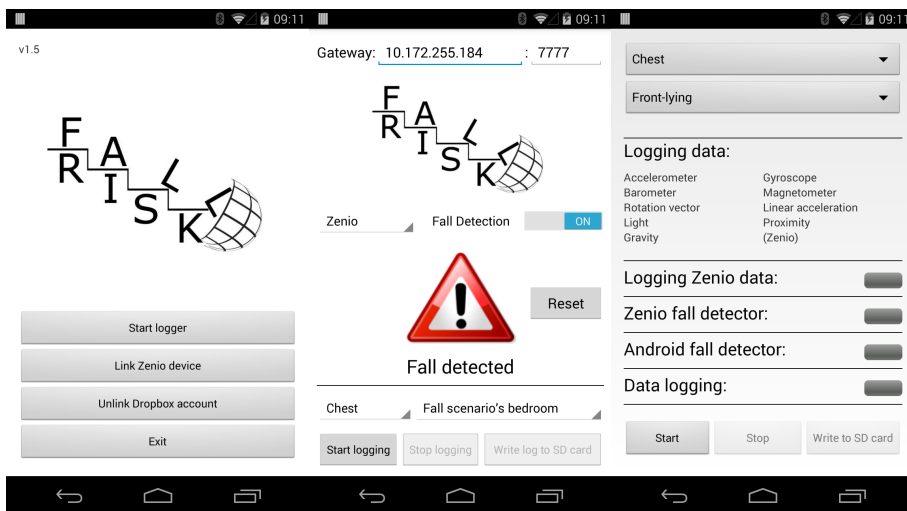
Figure 4: Detailed architecture of the OCarePlatform



Figure 5: Portable sensor testbed



Figure 6: The Zenio fall detector



(a) Fall detector start screen (b) View of the application when a fall was detected (c) Overview of the possible settings of the application

Figure 7: Screenshots of the smartphone application, used for detecting falls of an elderly

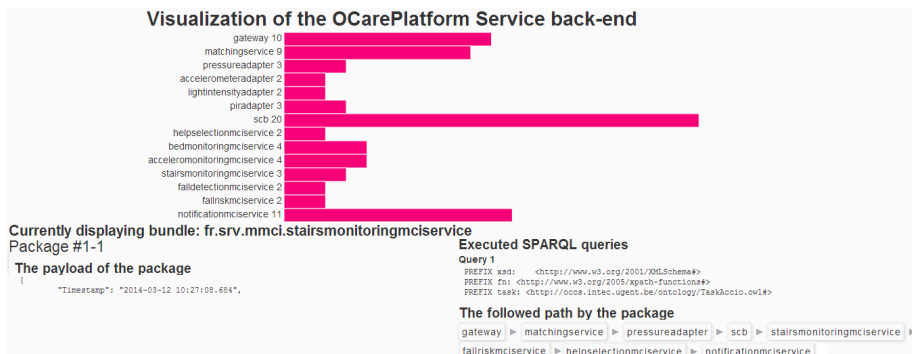


Figure 8: Example of the OCarePlatform visualizer

Legends - Tables

Table 1 - Monitoring sensors used for collecting contextual information

Legends - Figures

Figure 1 - Comparison between the current and envision fall risk and detection work processes

Today formal caregivers carry out standardized (TUG) tests to estimate the fall risk of an elder. Based on this test, advice is given (walking aids, physical exercises, adaptations in house). Follow-up is done through direct interaction with the elder. When an elderly falls, he/she has to use the PAS system to generate an alarm. This way, a desk operator is notified, who is capable of alerting an (in)formal caregiver. This is visualized in Figure 1(a).

In the envisioned process, as shown in Figure 1(b), a combination of sensor information with other contextual data can result in fall detection services with higher sensitivity. Add-on analysis of contextual data result in information enhanced alarm-services and high-level algorithms will dynamically forward detected events to the most adequate (in)formal caregiver. Feedback given by the caregiver after the event towards the system will allow for adaptive and learning intelligence in the service. The standardized test (TUG) is set as baseline and linked to the analysis of the Activities of Daily Living (ADL) behavior of the elder. When deviation of these patterns or non-optimum situations are detected, an early warning can be issued that a potential increased risk of falling occurs. The tailored intervention of experts and the early warnings will allow for better fall prevention.

Figure 2 - General, high-level, architecture of the FallRisk system

Information of sensors is sent for intermediate processing to the *Local Gateway*. This data is called Care Data (CD) and can be seen as raw data. The *Local Gateway* will enrich CD with information from the elderly and link it with other specific information, such as timestamps and identifiers. By this time, the information is called Meta Care Data (MCD). Next, the MCD will be sent to the Cloud. Data, originating from sensors of the smartphone or input

(in)formal caregivers entered through applications on the smartphone, are immediately transmitted as MCD to the Cloud. Within the Cloud, the *Controllers* will enrich MCD with Meta Care Concepts (MCC). MCC is a tag to identify the corresponding concept from the ontology. This way, the *OCarePlatform* is capable of interpreting the data in a more enhanced manner. MCD enriched with MCC is called Meta Care Information (MCI). MCI can be processed by the Cloud and by reasoning on MCI new knowledge can be generated. This new knowledge is called Meta Care Knowledge (MCK). When necessary, MCI and MCK can be sent to the smartphone in order to inform an (in)formal caregiver or the desktop operator.

Figure 3 - Prevalent concepts of the ambient-aware continuous care ontology for fall risk estimation and fall detection

This figure visualizes the prevalent concepts, relationships and properties of the domain model used by the *OCarePlatform*. The squares represent concepts, the dashed-dotted ones are concepts which actually contain subconcepts, but to give a clear overview, they were left out. The striped arrows model subclass relationships. The blue arrows model object properties, i.e., relationships between concepts. The utilized ontology is an extension of the ambient-aware continuous care ontology (ACCIO). The ACCIO ontology models knowledge and context needed to optimize continuous care processes using information technology. This ontology was extended with knowledge, which is needed to model the used contextual sensors and fall detection systems, detected falls, falls alarms, fall risk estimations and the relevant profile information of the (in)formal caregivers and elderly.

Figure 4 - Detailed architecture of the OCarePlatform

Information from the home is sent through a plethora of sensors to the *Local Gateway*. This gateway will send MCD to the *Controllers*. The *Controllers* is responsible for enriching the MCD with MCC and thus creating MCI and the management of the datasources, which keep track of the static information of the elderly and (in)formal caregivers. MCI is sent to the *Gateway Service* of the *OCarePlatform*. This service will forward the MCI to the *Matching Service*, which is capable of analyzing the MCI and which will send it to the

corresponding *Context Provider Service*. The *Context Provider Services* know which type of individuals need to be created and how they should be created by analyzing the axioms defined in the ontologies. Next, this ontological data is published onto the *Semantic Communication Bus (SCB)*.

The *SCB* uses an intelligent filtering system, by using core ontologies and filter rules, which is capable of sending only the useful data to the *MCI Services*. The *MCI Services* use a *Context Manager*, which contains (a subset of) the core ontologies used by the *SCB*, to specify the context they are interested in by defining filtering rules and registering them with the Context Disseminator. The *MCI Services* will generate new information in the form of MCK and sent this through the *Gateway Service* back to the *Controllers*. The *Controllers* will then notify the assigned (in)formal caregiver based on the information within the MCK.

Figure 5 - Portable sensor testbed

A portable embedded sensor network, emulating a smart home environment. This network has been integrated into a portable flight case.

Figure 6 - The Zenio fall detector

Figure 7 - Screenshots of the smartphone application, used for detecting falls of an elderly

Figure 8 - Example of the OCarePlatform visualizer

This figure shows the different component of the visualizer tool of the *OCarePlatform*. The visualizer tool graphically illustrates the reasoning process performed by the *OCarePlatform*, i.e., which *MCI Services* are triggered, which processing they each perform, how they are exchanged and which data is exchanged. At the top, the main component of the tool is shown. The different components/services of the *OCarePlatform* are visualized and the pink bars indicate how many MCI packets are currently residing within each of them. On the bottom left, a specific MCI packet residing within the *Stairs Monitoring MCI Service* is visualized. The bottom right, visualizes the processing, i.e., the SPARQL query, performed by this service and how the visualized MCI packet has flown through the different components/services of the *OCarePlatform*.

Summary

For elderly people fall incidents are life-changing events that lead to degradation or even loss of autonomy. Fall risk assessment is based on standardized tests carried out on long time intervals. Current fall detection systems are not integrated with other information technologies and often associated with undetected falls and/or false alarms. When a fall is detected, the systems employ a help desk that contacts the (in)formal caregivers in a pre-defined order, which leads to delays and distractions. Moreover, the desktop operator has no additional information on the situation the elderly is currently in and is not capable of assessing how severe the fall incident was. They also often lose time on calling the caregivers in the pre-defined order, time that could be very valuable.

In this paper, a social- and context-aware multi-sensor platform is presented, which integrates information gathered by a plethora of fall detection systems and sensors at the home of the elderly, by using a cloud-based solution, making use of ontologies. This integrated contextual information allows to automatically and continuously assess the fall risk of the elderly, to more accurately detect falls and identify false alarms and to automatically notify the appropriate caregiver, e.g., based on location or current task. The platform presented is capable of interacting with a wide range of heterogeneous sensors. Moreover, it is designed in such a way that intelligent, semantic techniques are used in order to deliver just-in-time information, about the fall risk and fall detection, to the elderly and their (in)formal caregivers.

It is important to note that presented platform does not need to be equipped with all the possible fall detection systems and available sensors to result in a reliable fall risk assessment and detection set-up. In order to evaluate the proposed platform, a proof of concept was implemented, making use of a portable sensor testbed and off-the-shelf sensors, which can easily be integrated within the platform. In addition, a visualizer was developed to visualise the complex dataflow within the cloud-based OCarePlatform. Both the portable sensor testbed and the visualizer make it possible to improve the platform by getting feedback from both the elderly and the caregivers by means of an interdisciplinary approach.