A wearable fall detection system based on Body Area Networks

Luigi La Blunda, Lorena Gutiérrez-Madroñal, Matthias F. Wagner, *Member, IEEE,* and Inmaculada Medina-Bulo *Member, IEEE*

Abstract—Falls can have serious consequences for people, which can lead, for example, to restrictions in mobility or, in the worst case, to traumatic based cases of death. To provide rapid assistance, a portable fall detection system has been developed which is capable of detecting fall situations and, if necessary, alerting the emergency services without any user interaction. The prototype was designed to facilitate a reliable fall detection and to classify several fall types. This solution represents a life-saving service for every person which would significantly improve assistance in case of fall events which are part of daily life. This paper will also introduce the fall analysis, which includes the generation of test events. To guarantee functional safety, the hazard analysis method System-Theoretic Accident Model and Processes (STAMP) will be applied.

Index Terms—Fall detection, Body Area Network, Wireless Smart Sensor Networks, Event Processing Language, STAMP

1 INTRODUCTION

FALL DETECTION is gaining in importance not only in aging societies, but also in the working society and in daily activities. According to the World Health Organization (WHO), fatal falls are estimated to be the second leading cause of accidental or unintentional death worldwide each year. People over 65 years suffer the most fatal falls [1]. In everyday life we are also confronted with risk of falling. Working in hazardous working conditions is another risk factor for causing fall events. An exemplary event could be a worker who falls during the night shift in the factory and no one is there to provide prompt assistance. Another example could be a technician which falls while maintaining isolated windmills. Considering these events the consequences can be fatal for the affected people. The WHO stated that annually 37.3 million fall events are severe enough to require medical treatment [1]. Fall events lead to the side effects of physical inactivity and loss of balance, especially among old people. Elderly people are scared to fall again and this uncertainty increases the risk of repeated falls. To counteract these life-threatening events fast assistance is necessary due to the fact that an unconscious person may not be able to call the emergency services. An approach could be the continuous tracing of medical and / or physical parameters via a wearable sensor network (see Figure 1).

A prototype in form of a belt has been developed which includes an electrocardiogram (ECG) harness and is based on a five sensor nodes Body Area Network (BAN). Each sensor node of the belt acquires continuously acceleration data including timestamp and the sensor node of the ECG harness provides the ECG signals including timestamp. In case of a fall the system should be capable to autonomously call the emergency services. Applying sensor fusion of physical and medical sensors we expect to improve the reliability of fall detection and possibly fall prevention. Another expectation is that the integration of medical sensors may facilitate the classification of different fall types. Detecting and classifying falls are critical in situations that may lead to loss of life if detected incorrectly. The BAN is generating a large amount of data in real time. Therefore data analysis technologies have to be used cf. [4]. Complex Event Processing (CEP) [5] is a data analysis technology which is used to manage and monitor in real time a huge volume of information that arrives in form of events with the lowest latency time. The CEP technology requires the usage of special software, the CEP engine. Each CEP technology provides a language called Event Process Languages (EPL). The EPL is used to detect relevant situations in real time by defining event patterns, in our case these relevant situations correspond to the detection of fall events. There are imperatives, rule-oriented and stream-oriented EPL types. For our research a stream-oriented EPL is used which will

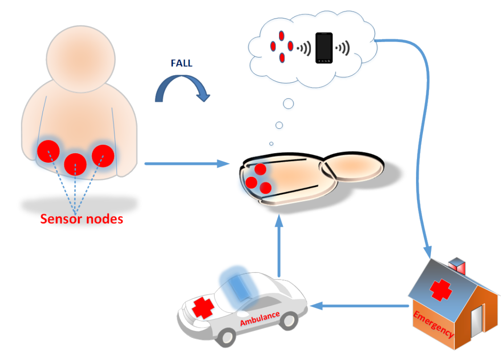


Fig. 1. Escalation scheme simulation [2], [3].

- *L. La Blunda and M. F. Wagner are with the Department of Computer Science and Engineering, WSN & IOT Research Group, Frankfurt University of Applied Sciences, 60318 Frankfurt am Main, Germany, E-mail: l.lablunda@fb2.fra-uas.de, mfwagner@fb2.fra-uas.de.*
- *L. Gutiérrez-Madroñal and I. Medina-Bulo are with the Department of Computer Science and Engineering, UCASE Software Engineering Research Group, University of Cádiz, 11519 Puerto Real, Spain, E-mail: lorena.gutierrez@uca.es, inmaculada.medina@uca.es.*

Manuscript received May 6, 2019.
(Corresponding author: Luigi La Blunda.)

be introduced in Subsection 3.2.

Since testing the system plays an essential role in verifying the reliability of our fall detection prototype by simulating all possible fall events, the IoT-TEG tool cf. [4], [6], [7] is used. IoT-TEG cf. [4], [6], [7] is capable to automatically generate test events of any event type, so it can be adapted to different types of event falls. With reference to the development of a reliable fall detection solution and the above mentioned expectations our ongoing research will be confronted with the following questions:

- Will the integration of medical sensors improve the reliability of the fall detection system?
- Can the system achieve a high level of acceptance among people?

The paper is structured as follows. Section 2 describes the related work regarding fall detection. In this section a short overview about existing solutions and different approaches for fall detection is given. Section 3 will describe the basic principles on which our ongoing fall detection research is based which contains the basic knowledge about falls, CEP [5] and IoT-TEG tool cf. [4], [6], [7]. In the successive section (Section 4) a detailed description of the fall detection prototype, including the generation of test events that includes the results of the fall analysis are introduced. Additionally, this section contains the usage of the ECG sensors and detected problems. Section 5 will introduce the hazard analysis method STAMP [8] which is applied on the fall detection prototype. Section 6 discusses the findings of the research questions of our ongoing research. Finally, in Section 7 a conclusion is given, including some indications for improvement which could be applied in the future work.

2 Related Work

In this section an overview about several fall detection approaches is given. There are several techniques which can be applied to detect fall events. Igual et al. [9] illustrate the following different types of fall detection systems:

- Context-aware systems
- Wearable systems

The functionality of context-aware systems depends on the environment, since the sensors and actuators must be installed in the living area (apartment, nursing home) in order to detect possible fall situations. The video-based context-aware solutions have the advantage to provide an accurate and reliable detection of falls with a fast assistance, but these systems have an issue regarding privacy. Patients using this solution are non-stop monitored limiting patient compliance. Additionally, the high purchase price is an obstacle for many patients and the dependency on the environment makes this approach useless in many application scenarios, because it would not detect fall events happening outside the networked area.

The other category of fall detection systems analyzed by Igual et al. [9] contains wearable solutions worn on the body and are based on a BAN. This solution is capable to provide a fall detection which is independent from the environment in contrast to context-aware systems. The analysis of Igual et al. [9] illustrates wearable fall detection systems using the sensor fusion with accelerometer and gyroscope data and built-in systems in form of smartphone sensors. For both categories of fall detection solutions (context-aware and wearable solution) several techniques were used. The following methods were applied for context-aware systems:

- Image processing and threshold-based recognition
- Image processing and classification models

The techniques used for the fall detection in wearable solutions are the following:

- Threshold-based approach
- Fall detection based on machine learning based data analysis

Taking the fact into consideration that it is an essential advantage to focus on a wearable solution independent of external infrastructure, the work of Li et al. [10] serves as an example cf. [4]. Their solution includes two wearable sensor nodes which are based on a BAN. These sensor nodes consist of an accelerometer and gyroscope and they are placed on the chest (Node A) and on the thigh (Node B, see Figure 2). The system distinguishes between two different

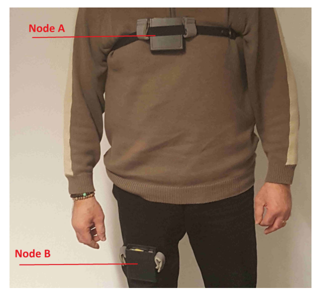


Fig. 2. System architecture according to Li et al. [10].

motion sequences which are used for activity categorization:

- Static postures:
  - Standing, sitting, lying
- Dynamic postures:
  - Activities of daily life (ADL) → walking, go up / down stairs, sit, jump, lay down, run
  - Fall like motions → quick sit-down upright, quick sit-down reclined
  - Flat surface falls → fall forward, fall backward, fall right, fall left
  - Inclined falls → fall on stairs

A 3-phase algorithm for fall detection is proposed by Li et al. [10]. The first phase of the fall detection algorithm examines if the person is in a static or dynamic position. If the analyzed position coincides with a static postures in the second phase it will be checked whether it corresponds to lying. If in lying position, it will be checked if the transition to this posture was intentional or unintentional (3^{rd} phase). To determine it, the previous 5 seconds of data is used. In the case it was unintentional the event will be classified as a fall. The proposed approach in [10] uses a threshold-based technique that is applied in the different phases of the algorithm. The weakness of this approach is to differentiate between activities of daily life and falling. Collado Villaverde et al. [11] propose a wearable fall detection solution based on a smartwatch using acceleration data in combination with machine learning techniques.

A different approach to detect fall events is presented by Lüder et al. [12] which apply an air pressure sensor in addition to the accelerometer. The hardware architecture proposed in [12] depicts a wearable solution which is worn on the hip and provides wireless data transmission via Bluetooth. To take meteorological disturbances into account Lüder et al. [12] incorporates an external barometric sensor as a reference. Another method for fall detection based on sensor fusion techniques is described by Gjoreski et al. [13]. Accelerometer and ECG data is used to detect fall events. The solution in [13] is capable of identifying person's movements and fall situations using wearable sensor nodes. These nodes are placed on the chest and thigh which is similar to Li et al's [10] approach. The advantage of the proposed solution by Gjoreski et al. [13] is the integration of medical sensors. The fusion of acceleration data and ECG signal facilitates the detection of anomalies in person's behavior and heart related problems that my lead to falls. According to their analysis differences in the ECG signal in different postures were detected. Lower beat rates in the static positions lying and sitting compared to walking were determined. Comparing the beat rates of both static postures (lying and sitting) differences were observed, where the beat rate of lying is lower than the one of sitting.

3 Background

3.1 Fall event analysis

The fall detection prototype is based on the approach proposed in [13], [14]. To detect a fall event a typical physical behavior is used which comprises the following phases:

- Prefall phase
- Falling phase
- Impact phase
- Postfall phase

The Figure 3 shows the mentioned phases. The prefall phase illustrates a stationary position of the person where the measured acceleration is around 1g ($9.81 m/s^2$). During the free falling phase the acceleration decreases close to 0g ($0 m/s^2$). Upon the impact, the acceleration reaches his maximum value for a short period of time. Subsequently, the acceleration is decreasing to around 1g ($9.81 m/s^2$) which represents the postfall lying phase. To apply this fall detection pattern the accelerometer data is used. For the impact detection the acceleration magnitude is used which is calculated with eq. 1 and to determine the body's orientation the single axis values of the accelerometer (α_x, α_y, α_z) are analyzed:

$$\alpha = \sqrt{\alpha_x^2 + \alpha_y^2 + \alpha_z^2} \tag{1}$$

Assuming that the person is in a standing position (static posture), the x-acceleration (α_x) corresponds to 1g ($9.81 m/s^2$) and the other two axis (α_y, α_z) would be close to 0g ($0 m/s^2$). As soon as the person changes his posture, the gravitational force will act on one of the other two axis (α_y, α_z) and the x-acceleration will be 0g ($0 m/s^2$). Combining

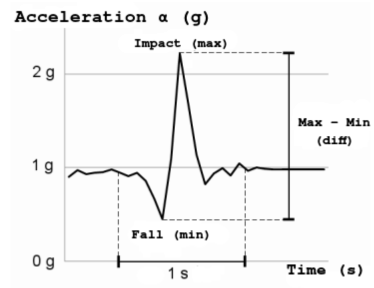


Fig. 3. Acceleration during impact according to Kozina et al. [14].

this information with the acceleration magnitude (see Formula 1) the system is able to detect the fall and additionally determine the type of fall. For applying this knowledge the EPL of EsperTech (Esper EPL from now on) [5] was used.

To ensure reliable fall detection, the system must be able to detect various types of falls. For this reason, the development of the fall detection prototype is also based on the creation of a test protocol covering different types of falls. Li et al. [10] and Pannurat et al. [15] are our references, as they not only represent possible solutions for fall detection, but also offer a versatile overview of possible fall scenarios.

3.2 Fall patterns based on Esper EPL cf. [4]

The EPL used is the Esper EPL [5], which is a streaming-oriented language and uses the CEP engine to execute the queries. The main reasons for its usage are the following:

- The syntax is based on SQL → complex events can be easily formulated.
- It can be embedded in Java applications.
- It is open-source.
- CEP engine of EsperTech processes around 500.000 events per second on a workstation and between 70.000 and 200.000 events per second on a notebook (according to its developer) → This is an essential feature to simulate time-critical applications.

In comparison to SQL, which is based on tables, EPL works on the continuously incoming data stream. Therefore, a row from a SQL table corresponds to an event in the event stream.

To define rules in CEP the incoming event should be characterized in detail to specify incoming data for the CEP-engine. After the event creation, rules, as well called event patterns, should be determined to categorize the incoming input in fall events or daily activities.

Example 1. Fall pattern based on Kozina et al. cf. [4], [14].

```
//1. Definition of the event - incoming data for CEP
*---------------------------------------------*
create schema BodyEvent(PersonID integer,accelS1
    double,accelS2 double,timestamp string)

//2. Definition of Event pattern
*---------------------------------------------*
select a1.accelS1, a2.accelS1, a1.accelS2, a2.
    accelS2 from pattern [every(a1=BodyEvent(a1.
    accelS1 <= 9.81) -> a2=BodyEvent(a2.accelS1-a1.
    accelS1 >= 9.81 and a1.PersonID = a2.PersonID)
    where timer:within(1sec)) or every (a1=BodyEvent
    (a1.accelS2 <= 9.81)-> a2=BodyEvent(a2.accelS2-
```

```
a1.accelS2 >= 9.81 and a1.PersonID = a2.PersonID
) where timer:within(1sec))];
```

The illustrated EPL query (see Example 1) is based on the physical principle shown in Figure 3 (see Subsection 3.1). It should be highlighted, that two nodes were used for this EPL query (one frontal node and one lateral node) to apply the fall detection, but in future this query will be extended for all the sensor nodes of our prototype BAN. The four nodes BAN architecture is currently only used for redundancy purposes. In the Example 1 the following event properties are used for the definition of the event pattern:

- a1.accelS1 → initial acceleration value of sensor node 1.
- a2.accelS1 → successive acceleration value of sensor node 1.
- a1.accelS2 → initial acceleration value of sensor node 2.
- a2.accelS2 → successive acceleration value of sensor node 2.

Referring to the selected properties this example checks, if the initial acceleration of node 1 (a1.accelS1) is $<= 9.81$ m/s^2 which indicates that the person is in a stationary position. Subsequently, it will be checked if the difference of the successive acceleration (a2.accelS1) and the first acceleration (a1.accelS1) within 1 second is $>= 9.81$ m/s^2. If this condition is fulfilled, it is an indication that the person has suffered an impact. Adding the OR disjunction the second sensor node can be integrated and the statement is able to detect a fall in case one of the node's data matches the Esper EPL query and the values of the acceleration correspond to the same person.

3.3 IoT test event generator cf. [4]

IoT-TEG [6], [7] is a Java-based tool which takes an event type definition file and a desired output format JSON, CSV and XML, the most common across IoT platforms. IoT-TEG is made up of a *validator* and an *event generator* (Figure 4). The validator ensures the definition follows the rules set by IoT-TEG. The generator takes the definition and generates the indicated number of events according to it.

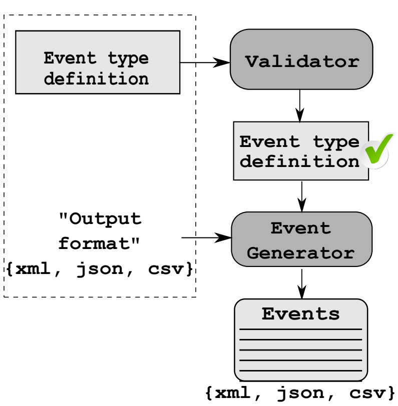


Fig. 4. IoT-TEG Architecture cf. [4], [6], [7].

Previous studies suggested there were no differences in testing effectiveness between using events generated by IoT-TEG, or events recorded from various case studies [6], [7]. Moreover, thanks to its implementation, IoT-TEG can be used to do different type of test: functional, negative, integration, stress, etc; indeed an example of its usability can be found in [7], [16], where IoT-TEG has been used to apply mutation testing [17]. These results confirm IoT-TEG can simulate many types of events occurring in any type of applications, it can do different type of tests and it can solve the main challenges developers face when they test event-processing programs: lack of data for testing, needing specific values for the events, and needing the source to generate the events. For the sake of clarity, Example 2 shows an event type definition.

Example 2. Event type definition example cf. [4].

```
<?xml version="1.0" encoding="UTF-8"?>
<event_type name="TemperatureEvent">
<block name="feeds" repeat="150">
<field name="created_at" quotes="true" type="
    ComplexType">
 <attribute type="Date" format="MM-dd"></attribute>
 <attribute type="String" format="T"></attribute>
 <attribute type="Time" format="hh:mm"></attribute>
</field>
<field name="entry_id" quotes="false" type="Integer"
     min="0" max="10000"></field>
<field name="temp" quotes="false" type="Float" min="
    0" max="500" precision="1"></field>
</block>
</event_type>
```

The defined event type contains three properties: `created_at`, `entry_id` and `temp`. These properties are defined as fields in the event type definition. The `created_at` field is complex type and `entry_id` and `temp` are simple types.

Apart from the mentioned challenges that IoT-TEG solves in order to test event-processing programs, it incorporates a specific functionality for testing programs that use the Esper EPL [5]. This functionality helps to automatically generate events with specific values in accordance with the program which will process them. IoT-TEG analyses the Esper EPL queries and generates events depending on the logical and relational operations.

4 Fall Detection System Prototype

4.1 Architecture

A first prototype was described in [2], [3], cf. [4]. This paper presents an improved version of the prototype in [2], [3], cf. [4], which includes essential developments that contribute to reliable operation of the system. Additionally, the improved hardware design meets the requirement for patient compliance. A more user friendly design facilitates the freedom of movement for the user. As a safety critical system for medical purposes, a redundant hardware design protects the system against a total system failure which would have severe consequences in case of fall events. Referring to Jämsä et al. [18] the best approach is to position the accelerometer near to the waist. Taking into consideration these aspects a wearable belt solution was developed which is based on a four sensor nodes BAN (see Figure 5). The four sensor nodes (S1 - S4) positioned on the belt are acting as peripherals and are continuously acquiring acceleration data which is sent via Bluetooth Low Energy (BLE) to the smartphone (central device).

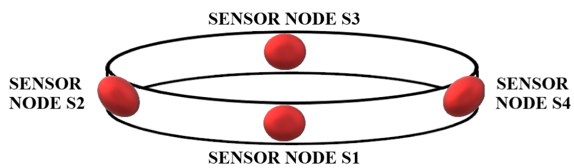


Fig. 5. Four sensor nodes BAN - Belt [2], [3].

The dataset consists of the following information:

$$\text{SensorID}, \alpha_x, \alpha_y, \alpha_z \text{ measured in m}/s^2$$

- SensorID $\rightarrow$ sensor node identification.
- $\alpha_x \rightarrow$ acceleration value in X-direction.
- $\alpha_y \rightarrow$ acceleration value in Y-direction.
- $\alpha_z \rightarrow$ acceleration value in Z-direction.

The central device receives the incoming sensor data which will be stored in separate data files to evaluate the event. The belt solution reflects the above criteria of a safety critical systems. The proposed architecture (see Figure 5) is based on a mirroring principle of the opposite sensor nodes which provide identical acceleration values, only with different signs. If a sensor node fails during operation, the opposite node can be used as a reference to ensure accurate evaluation of the event. Considering the following scheme (see Figure 6) this fall detection belt solution facilitates the recognition of different fall event types. The positioning of the nodes around the hip allows a precise fall characterization.

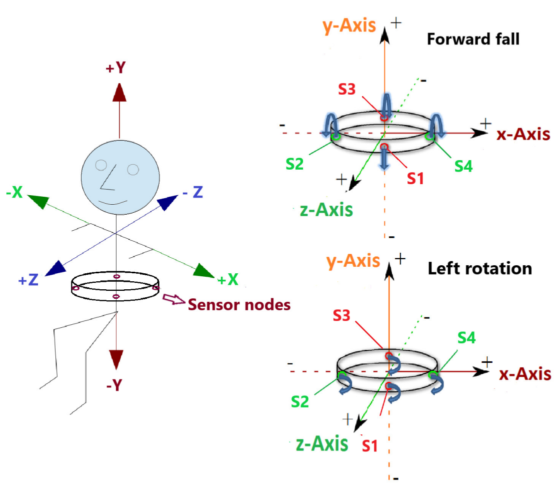


Fig. 6. Three axis reference scheme [2], [19].

Fall type example: a person suffering a multi phase frontal fall because of a fainting. Analyzing the fall event, it can be determined that the fall caused originally by fainting leads firstly to a left rotation of the person's body (prefall phase). In the next phase the forward fall results in the person's impact to the ground. Projecting this fall event on the reference scheme (see Figure 6), the body's orientation (e.g. left rotation) can be determined on the basis of the individual acceleration values (α_x, α_y, α_z). Additionally, the acceleration magnitude (see Formula 1) can be used for the detection of the impact which is categorized by a temporary increase in the acceleration, as described in the previous Section 3. Using the Esper EPL query (Example 1) stated in the Subsection 3.2 the system is capable to detect a general fall event based on the physical principle proposed by [14]. Taking into consideration the test protocol stated in Section 3 typical fall events (e.g. rolling out of the bed) in nursing homes and hospitals can be detected by this solution. In the subsequent subsection, the behavioral analysis of the acceleration values during one of the mentioned typical fall events is presented, the *fall against wall* (FAW). Additionally, test events are generated to simulate the FAW fall type.

4.2 Fall simulation test events

The fall to generate the test events consists on the impact of the person with a wall and falling on the knees and then on the chest: FAW. In this study we have used two healthy subjects, and we have recorded falls with all possible realism while also trying to avoid risks. They have been doing FAW fall test for a period of 2 minutes. The analyzed data and videos can be found in [20]. In this analysis the following steps have been done:

4.2.1 Study of the values cf. [4]

Given that the sensor (S1) is the one that suffers the impact, its acceleration values are the first to be analyzed. The goal is to study the acceleration behavior during a fall in order to generate test events, so the acceleration values are normalized ($N(m/s^2)$). After the normalization the impact, peaks, have to be detected; we have considered a peak when the normalized acceleration is greater than 0,7 ($N(m/s^2) > 0,7$).

While performing the data analysis, it has to be taken into account that the values suffer alterations because several factors: the person movement, the person bounces against something (floor, wall, etc), the collocation of the sensors to the original position after a fall, sensor pressure because an impact or the person is laying over it, etc.

After applying the previous rule in all the fall data and taking into account the alterations because the mentioned factors, the impacts are detected.

4.2.2 Fall identification and analysis

Once the peaks are detected, a range of values, including the peaks, are selected in order to analyse data properly and to study the acceleration behavior during FAW fall. The range of extracted values is a set of data that happens in less than a time window of 1 second, to meet the fall rule of [12] described in eq. 1, see Section 3.

Figure 7 confronts the normalized acceleration behavior during the previous FAW falls of person 1 and person 2. These comparisons show a similar behavior of the acceleration during the FAW fall.

We have decided to define the acceleration behavior with normalized values; so the normalized acceleration behavior during the FAW consists on:

1) The variation of its values while the person is walking. We have divided this rule in two rules:
   a) The normalized acceleration values go increasing in a range [0, 0.35].
   b) The normalized acceleration values go decreasing in a range [0, 0.35].

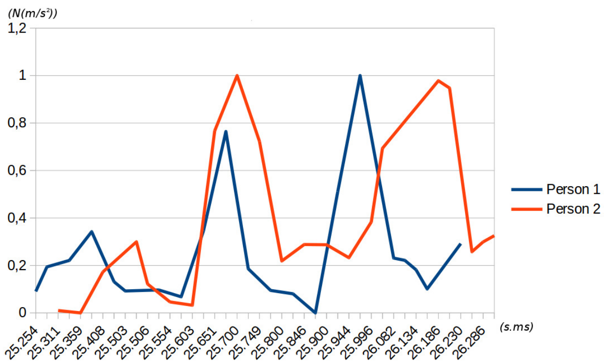


Fig. 7. Acceleration comparison during FAW fall cf. [4].

2) As a consequence of the impact of the person against a wall, the acceleration normalized value has to be greater than 0.7.
3) The normalized acceleration values decrease to a range [0, 0.35]. The values of the acceleration are in the mentioned range depending on the size of the person; the larger person results in a longer range and if the person retains a position prior to a fall thanks to the wall. Moreover, a subtle peak could appear as a consequence of a rebound.
4) A second impact happens when the person hit the ground, the acceleration normalized values have to be greater than 0.7.
5) Finally, the person is laying on the ground and the normalized acceleration value decreases. The values of the acceleration are between [0.10, 0.35] and no subtle peaks appear.

The same analysis process has been done to the rest of sensors, and the acceleration follows the same pattern.

4.2.3 To define the fall event

Once the fall acceleration behavior has been observed, the next step is to define the fall event in order to generate test events with IoT-TEG cf. [4], [6], [7]. As it was explained in Subsection 3.3, the event type attributes have to be defined using the `<field>` element. The fall event contains one attribute, the acceleration, which is float, `type=``Float''`, and its values are not quoted, `quotes=``false''`. A new parameter in IoT-TEG has been defined as a consequence of the previous fall study. Given that the acceleration values follow a specific behavior, it is necessary to include the `custom_behavior` property in the `<field>` element to define the behavior of any event attribute; in this study, the acceleration. In the `custom_behavior` property the path to the file that includes the behavior of the event attribute has to be written. The Example 3 shows the complete fall event definition (FallEventType).

Example 3. Fall event type definition, cf. [4].
```xml
<?xml version="1.0" encoding="UTF-8"?>
<event name="FallEventType">
<block name="feeds" repeat="100">
<field name="acceleration" quotes="false" type="
    Float" custom_behavior="/Path/To/Rule/File">
</field>
</block>
</event>
```

IoT-TEG includes a new functionality, which has been implemented to simulate the desired behavior of an event attribute with a `custom_behavior` property in its event type definition. This functionality allows to generate values of the event attribute following a behavior that the user has described in a file. In order to explain how the user has to define the desired behavior of an event attribute, we are going to use the FAW fall behavior rules (see Example 4). In a XML file the number of simulations has to be indicated, the events involved in a simulation will be calculated according to the total number of events to generate and the desired simulations. For example, if the number of test events to generate is 100, number indicated in the event type definition file, `repeat="100"`, and the number of desired simulations is 5, `simulations="5"`, number indicated in the behavior rules definition file, the number of events to simulate the behavior is 20. In Example 4, the user ask to generate 100 test events and 5 falls (simulations), so 20 events will be used to define a FAW fall.

Variables can be defined if they are needed in the behavior rules. They can be defined in the file where the behavior rules are included using the `<variables>` tags. To define them a name and a value have to be given to the variables. The value can be defined as a fixed value with the `value` property, or using a range with the `min` and `max` properties. Moreover, in some variables are involved in the value of another variables; this is indicated using the variable with an specific format `$(variable)`, see Example 4. In addition, arithmetic operations can be done in the definition of the variable values. Let us see how using them in the FAW fall example. To define the acceleration behavior during a FAW fall three variables are defined: `Base`, `ImpactWall` and `Impact`. Given that the acceleration behavior during a FAW fall has being done according to the normalized values, the variables and rules have been defined according to that analysis. The acceleration value in a stationary position is variable depending on the person, so we have considered the established value, $1g \approx 9.81 m/s^2$. That is the fixed value of the `Base` variable. To determine the values of the impacts, we have taken into account that the normalized value have to be $> 0,7$. So, to ensure that the impacts, `ImpactWall` and `Impact`, have a value meeting the mentioned condition the minimum value of the impact is the sum of the `Base` and `Base` multiply by `Base`$*0,7$; and the maximum value of the impacts is $3g \approx 9,81 * 3 =$`Base`$*3$.

Once the variables are defined, the rules have to be determined. The first step is to indicate the `weight` for each rule in order to calculate the number of events to generate for each rule for each simulation. Following the simulation values the number of events to generate for each rule, according to the assigned weights, is: $20 * 0,25 = 5$ events will be generated for the first rule, another 5 events for the second rule, 1 event for the third rule, 5 events for the fourth rule, 1 event for the fifth rule and the remaining events, three events, for the sixth rule. We have to assign zero to the weight `weight="0"` to indicate how the remaining events have to be generated.

To define the rules, `min`, `max` and `value` properties can be used as well as the arithmetic operations and the references to another variables. Moreover, the `sequence` property can be used to obtain values lower or higher than the one generated previously: `inc`, to increase the value, or `dec`, to decrease the value.

Example 4. Rules to define a FAW fall, cf. [4].
```
<?xml version="1.0" encoding="UTF-8"?>
<custom_conditions simulations="5">
<variables>
<variable name="Base" value="9.81"/>
<variable name="ImpactWall" min="$(Base)+$(Base)*0.7
    " max="$(Base)*3"/>
<variable name="Impact" min="$(Base)+$(Base)*0.7"
    max="$(Base)*3"/>
</variables>
<rules>
<rule weight="0.25" min="0" max="$(ImpactWall)*0.35"
     sequence="inc"/>
<rule weight="0.25" min="0" max="$(ImpactWall)*0.35"
     sequence="dec"/>
<rule weight="1" value="$(ImpactWall)"/>
<rule weight="0.25" min="0" max="$(Impact)*0.35"/>
<rule weight="1" value="$(Impact)"/>
<rule weight="0" min="$(Base)+$(Base)*0.10"
max="$(Impact)*0.35"/>
</rules>
</custom_conditions>
```

Thanks to the included properties and parameters in the IoT-TEG new functionality, the desired behavior rules that follow the normalized acceleration values can be defined. Given that we have considered to define the FAW fall behavior rules according to the normalized values, the values to generate depend on the maximum value. Due to there are two values that can be the highest one, `ImpactWall` or `Impact`, the rules that depend on the maximum value contains the reference to the value, `ImpactWall` or `Impact`, according to the proximity of the rule. For instance, the first and second FAW fall rules contain a reference to `ImpactWall`, the impact in the wall (third rule), which happens after the person is walking, something described in the first and the second rules. The fourth and sixth rules contain a reference to `Impact`, the impact on the ground (fifth rule), which happen after the person is falling and the person is laying on the floor, fourth and sixth rules.

It is needed to highlight that to obtain these rules to define the behavior of the normalized acceleration several test have been done. Once we obtained the desired results, test events were generated as they were necessary. The Figure 8 shows the acceleration values of some of the generated FAW falls using IoT-TEG and the new functionality.

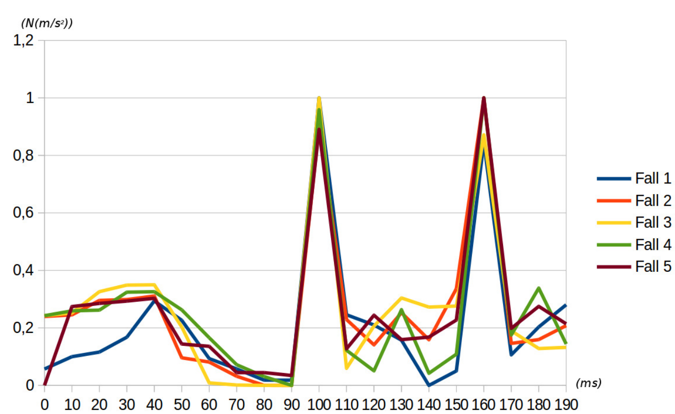


Fig. 8. IoT-TEG generated FAW falls cf. [4].

4.3 Sensor fusion

Analyzing the literature regarding fall detection, the existing solutions show certain weaknesses in the reliable detection of falls [9], [10], [12], [15], [18]. Considering the approach of Gjoreski et al. [13], the results with the fusion of physical sensors and the ECG sensor increase reliability. Especially the evaluation of the ECG signal leads to an essential improvement of the system's accuracy. According to [13] the ECG signal could be used to distinguish between different postures. The inclusion of medical parameters could even provide a fall prediction that would represent a significant progress in health care and prevent fatal injuries. Based on these results the proposed fall detection belt has been upgraded with a portable 3-channel ECG sensor. The successive illustration depicts the BAN structure of the update prototype architecture.

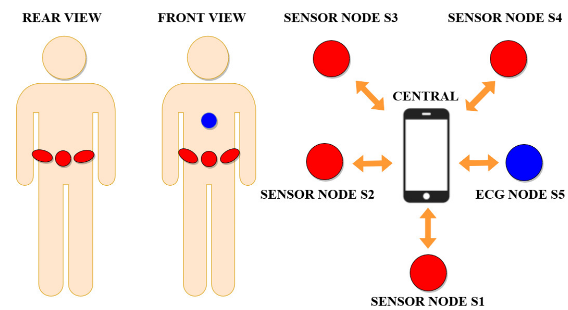


Fig. 9. ECG BAN.

The updated BAN includes four sensor nodes placed on the belt (S1 - S4) and an additional node on the chest (S5). These five nodes are acting as peripherals and they are continuously providing sensor data to the smartphone (central) via BLE. The sensor nodes S1 to S4 send the acceleration data and the ECG sensor sends ECG signals to the central device. Fusing this information the central device analyzes the events for possible fall events.

Before the ECG sensor can be fully integrated into the prototype, some tasks had be solved. The first task to solve is a continuous recording of the ECG during the person's daily activities. When the ECG electrodes lose contact with the skin surface because of movement, this leads to increased noise in the signal. Therefore, a solution for reliable ECG recording have to be developed. The second task is to perform a validation of the ECG sensor. For this purpose, the ECG measurements must be compared with measurements taken by a clinical ECG device. Based on Gjoreski et al. [13] the ECG signal can be relevant to detect fall events as mentioned in Section 2. Based on this, another task will be the determination of ECG patterns that provide relevant information for fall events.

The first test measurements with the ECG sensor confirmed the noisy and unstable ECG signal during movements. To ensure a stable and continuous signal during daily activities, an adjustable and flexible ECG harness has been developed with prefabricated electrodes positioning and the ability to adapt to any body shape (see Figure 10).

After completing the development of the ECG harness the measurements were repeated using the harness solution. A more stable signal resulted, especially during walking

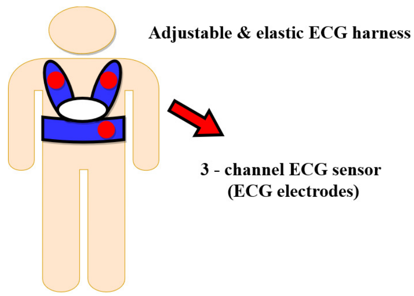


Fig. 10. ECG harness.

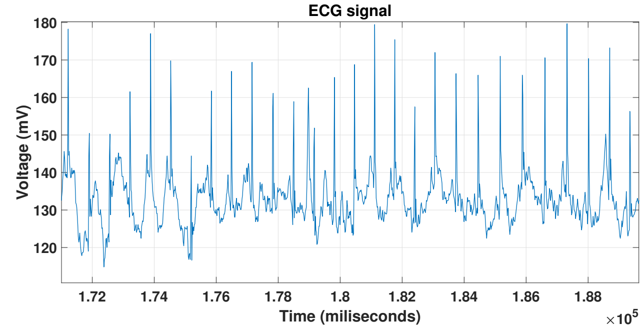


Fig. 11. Baseline shifting ECG signal [23].

and the sitting procedure the stability of the ECG signal was improved significantly. The artifacts in an ECG signal which result during motion may be used as a recognition pattern for fall events. After the positive results regarding the improvements of the signal's stability, validation tests were conducted in a medical lab with the assistance of a physician. In clinical settings ECG devices with 10 electrodes are being used but for wearable or emergency purposes (e.g. ambulance) a 3-Channel ECG device is recommended because it provides less wiring and satisfies the aspect of patient compliance [21]. According to a study by Antonicelli et al. [22], the use of a 3-lead ECG may be essential to avoid delayed treatment of specific heart diseases, such as elderly people who suffer from chronic heart disease and need continuous ECG monitoring. In addition, the analysis in [22] led to the result that a 3-lead ECG provides qualitatively similar evaluation as a 12-lead ECG. Comparing the measurements taken in the medical lab with the ECG measurements provided by our ECG sensor the correct functionality of our sensor could be confirmed. Based on this state of knowledge, ECG measurements are performed during the fall simulations based on [10] and [15] as part of a master thesis [23]. The aim of this work has been to evaluate the ECG patterns for essential artifacts during the fall and to apply machine learning methods to improve the system's ability to detect fall events and, if possible, predict fall events with the additional information of the ECG sensor. The application of machine learning techniques based on accelerometer data is also being investigated.

4.4 Detected problems

After testing the used fall detection prototype, some problems were found. Moreover, some considerations will be applied in future tests.

First of all, we are going to explain the problems related to the current prototype. The synchronization in the prototype is an issue. There is a lack of synchronization not only in the amount of data, but also in the timestamp. Some sensors transmit more data than the others. The four sensors were working during the FAW fall test, but the obtained acceleration values were from three of them, one of the sensors did not transmit data in one period of the test.

A hardware issue is the durability of the battery, which is 2 to 3 weeks depending on the use. If we want to use this system in everyday life, we have to make sure that the battery life is extended. If we consider the used ECG sensor architecture, we encountered differences in signal quality that includes noise and baseline shifting (see Figure 11).

The reasons for these distinctions in signal quality were explained by the supervising physician. Baseline wander (BW) disturbances are caused by variations in electrode-skin impedance, patient's movement and breathing during the ECG measurements. Muscle contraction is common for people which suffer from tremor or fearing the ECG measurement and for disabled people. Certified clinical devices have an integrated filter which can be applied to smooth the signal [21], [24]. To solve this problem the application of a low and high pass filter in our setup is considered [23].

Regarding the analysis of the acceleration values, it was observed that some values could lead to misinterpretation. The reason was that the test persons quickly got up after the fall simulation instead of lying on the ground. Since not only the measured values but also the video recordings of the fall simulations were analyzed for a profound analysis, the problem could be determined. In a real situation, if a person falls and is able to stand up quickly it means that the person is conscious and able to move. On the contrary, if the person falls and does not get up after a while, it means that the person may be unconscious or unable to make the emergency call. Therefore, waiting at least 10 seconds in our test scenarios is helpful to perform an accurate fall analysis during testing.

5 Example Application of STAMP as Hazard Analysis Method

5.1 Introducing STAMP

A fall detection system is a safety critical system requiring certification according to a safety standard (e.g. IEC 60601-1-11) [25]. To satisfy functional safety requirements it is fundamental to apply hazard analysis methods during all development phases and operation to analyze the behavior of the system in case of malfunctioning [26]. In the following STAMP will be applied in some architectural parts of the fall detection system for functional safety validation.

STAMP is a comprehensive hazard analysis method based on system theory. A major goal of this approach is to control or eliminate hazards. This approach proposed by Leveson [8] is used to identify probable accident causes that are categorized as follows:

- Accidents based on hardware and software component failure
- Unsafe interactions among components

- Complex human machine interfaces and human error models
- Errors in system design
- Faulty requirements

STAMP treats accidents as a control problem. To prevent accidents safety constraints derived from system safety requirements must be enforced inside the different system hierarchy levels. If the given safety constraints are violated accidents may occur. Checking the enforcement of safety constraints leads to an additional layer of system testing.

5.2 STAMP - Hazard analysis

STAMP begins on the system level and proceeds top down into the system hierarchy. The hierarchical-based analysis in STAMP is based on control loop structures which are iterated from the high abstract level (overall system view) to the lower system levels to build up hierarchical system models [8]. The controller unit of the control structure is acting as a master and includes a process model to determine control actions that are executed by the actuators to control the defined process (controlled process). A feedback loop (measured variables) is provided by the sensor unit which informs the controller about the process state. This information is used by the master to initiate a control action (controlled variables) to run the system within predefined limits.

STAMP procedures:

- *Definition of accidents:* Person has fallen and the fall was not detected by the system.
- *Definition of hazards:* Data from 1 to 4 sensors of the belt are not sufficiently correlated in time (synchronization error).
- *Definition of safety requirements and constraints:* The safety requirements and constraints are based on the hazard which causes the accident. Safety constraints are used to describe non-permissible system operations in order to ensure safe operating conditions. A top level safety requirement is the real time detection of fall events. The timestamps of all four nodes must be synchronized. If constraints are violated system migrates to unsafe or hazardous state.
- *Definition of safety control structure:* A high-level safety control structure should be defined which contains the system's components and the process which should be controlled. The control structure below (see Figure. 12) depicts the control loop of the controlled process, the movement of the person. The sensors provide sensor data to the controller (smartphone application). The smartphone contains a data acquisition algorithm to collect the incoming data and a fall detection algorithm to analyze this data. If the event corresponds to a fall, the alert system on the mobile device will call the emergency services for intervention. The actuator part of the system contains the control of the data acquisition process, i.e. timers, sleep mode etc.

The lower level control loops (see Figures 13 and 14) illustrate a detailed internal control structure of the controlled processes. These processes are interacting as a controller in the lower level structures. Taking into consideration the second level safety control loop, the movement process control (controller) receives the sensor data. Based on the data a control command will be executed to control the BAN which sends a feedback to the movement process controller. The third level control loop zooms into the BAN control, which is the master (controller) that actuates the wake up procedure of the sensor nodes to control the nodes. These will deliver sensor data, including the time to the BAN control.

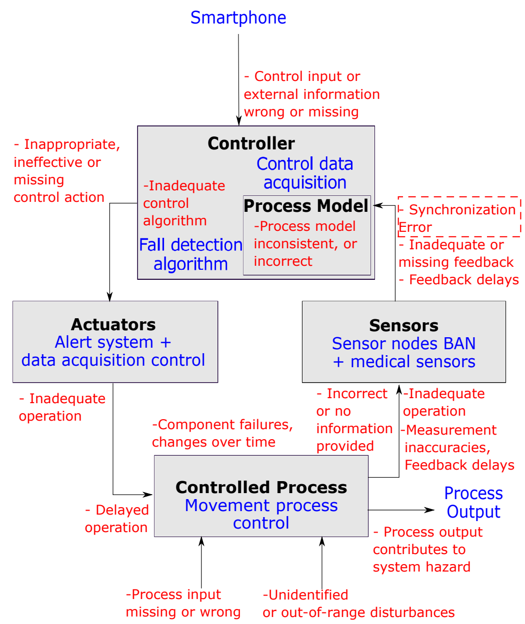


Fig. 12. System Safety Control Structure - Hierarchy level 1 → Black label (system components), blue label (description of system components), red label (possible hazards).

The lower level control loops (see Figures 13 and 14) illustrate a detailed internal control structure of the controlled processes. These processes are interacting as a controller in the lower level structures. Taking into consideration the second level safety control loop, the movement process control (controller) receives the sensor data. Based on the data a control command will be executed to control the BAN which sends a feedback to the movement process controller. The third level control loop zooms into the BAN control, which is the master (controller) that actuates the wake up procedure of the sensor nodes to control the nodes. These will deliver sensor data, including the time to the BAN control.

To provide a more detailed hazard analysis two more control loop layers were created which are illustrated in the successive illustration (see Figure 14). The fourth level of the system's control structure zooms into Sensor node control (controlled process in Figure 13) which is the control unit (controller). Considering this control layer the controller receives the sensor's services and characteristics which are provided via BLE and contain the sensor values. The sensor node control initiates a scanning command (actuator) to detect the sensor properties which are provided by the BLE interface (controlled process: Bluetooth sensor detection control).

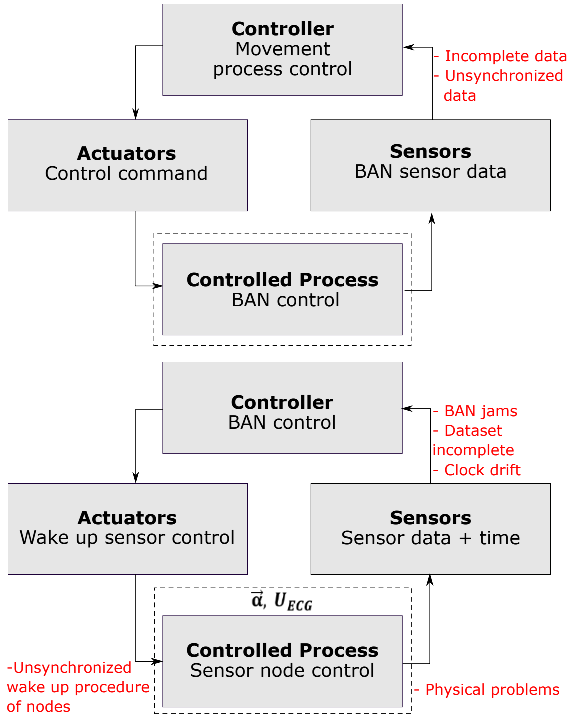


Fig. 13. Safety Control Structure - Hierarchy level 2 & 3.

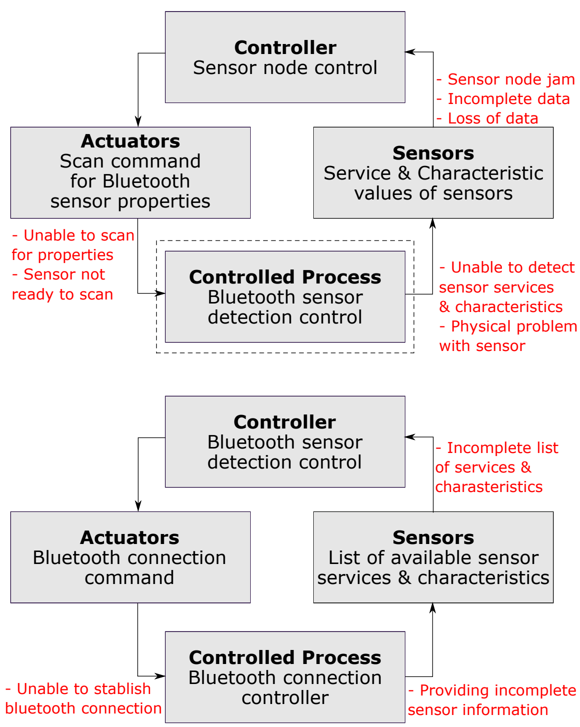


Fig. 14. Safety Control Structure - Hierarchy level 4 & 5.

Taking into consideration the fifth level of the control loop, which is depicted in Figure 14, an accurate view of the controlled process Bluetooth sensor detection control in level 4 is provided. The master Bluetooth sensor detection control (controller) actuates a connection command via BLE (actuator) to establish a connection to the sensors (controlled process: Bluetooth connection control) which sends a list of available sensor's services and characteristics to the controller.

All these hazards (see red labels in Figure 12, 13 and 14) lead to safety constraints. Some of the safety constraints are defined as follows when considering the control loop structures of the second and third system levels:

- Second level control structure $\rightarrow$ to avoid data loss and provide a reliable evaluation of events by the movement process controller, the sensors should provide synchronized data.
- Third level control structure $\rightarrow$ a synchronous wake up procedure of all sensors in the BAN must be ensured to avoid data loss.

If the control structures of the fourth and fifth system levels are considered, some of the safety constraints are defined as follows:

- Fourth level control structure $\rightarrow$ It must be ensured that the sensor information (services & characteristics) is transmitted completely.
- Fifth level control structure $\rightarrow$ A reliable and stable BLE connection to all nodes of the BAN must be established.

Violating one of the above-mentioned safety constraints, a chain reaction of hazards (non-permissible operations) is triggered in all system layers, which leads to malfunction of the system. The following scenario, which reflects a possible chain reaction of hazards, shows the possible effects on the system functionality. The incomplete list of sensor's services and characteristics (control loop - lower level 5) and sensor node jam (control loop - lower level 4) can lead to clock drift (control loop - lower level 3) and unsynchronized data (control loop - lower level 2) in the upper levels. Merging all the hazards from the lower levels may cause the synchronization error (see description in subsection 4.4) in the top level control structure (see Figure 12) and lead to violation of the safety constraint $\Delta t <= 50$ms between the timestamps of the sensor nodes. The combination of these possible hazards makes the system inoperable to detect falls in real time.

It is important to emphasize that this is only the beginning of a complete STAMP analysis, which represents a fraction of the system. We will use it for all parts of the complete architecture.

6 Findings with Respect to the Research Questions

Our ongoing research solved part of the research questions (see Section 1). The following subsections relate our results to the corresponding research questions.

6.1 Will the integration of medical sensors improve the reliability of the fall detection system?

Considering the reliability in relation to the integration of medical sensors in our fall detection system, by analyzing the literature [13] it can be concluded that medical parameters significantly enhance the capabilities of the fall detection. Consultation with physicians [21] confirmed that the data acquisition of medical parameters in parallel to the acceleration data cover a fall event initiated by medical conditions comprehensively. Our ongoing tests performed with the ECG sensor harness indicate that the ECG provides relevant information for an accurate fall detection. Current work analyzes disturbances of the ECG signal and explores the use of machine learning methods to detect relevant normal and abnormal medical patterns [23].

6.2 Can the system achieve a high level of acceptance among people?

According to the feedback from our test subjects, our system has achieved a high level of acceptance:

- The hardware architecture is flexible and adaptable to any body shape.
- The current prototype has decreased in size.
- With the belt and harness solution, the system is not visible from the outside and can be worn comfortably. This aspect increases patient compliance.

7 Conclusion

This proposed portable fall detection system aims to provide rapid and efficient assistance to people who are in life-threatening situations due to falls.

The test measurements performed with the test subjects resulted positively regarding the detection of fall events and the acceptance of wearing such a system. Incorporating the ECG sensor proved the effectiveness of Gjoreski et al's concept [13] combining physical and medical sensor information (acceleration values & ECG) to ensure more accurate detection of falls and, if necessary, fall prediction.

Using the IoT-TEG tool [6], [7] facilitates the generation of events to recognize falls which are based on [14]. We have the ability to assign behavior rules to as many event attributes as the event type contains, and the event attribute values follow the specified behavior. IoT-TEG [6], [7] has the ability to adapt the behavior of the analyzed event attribute, because it was designed to generate events of any event type to test systems which manage events. In the further development process it is planned to use machine learning and complementary CEP in order to use certain ECG patterns in combination with the acceleration data. These should enhance a reliable detection of falls. Since the IoT-TEG tool [6], [7] is in constant development to ensure an accurate generation of events, further functionalities will be integrated in the future to meet all test requirements.

However, first the synchronization problem of the BAN must be solved, which can compromise the system's functionality. A new hardware platform will be used, which contains a real time operating system, which facilitates the synchronization of different tasks in a wireless sensor network. In addition, the new microcontroller should also meet other requirements of patient compliance.

Referring to the research questions (see Section 1), several aspects have been solved (see Section 6). Due to the complex nature of the problem, ongoing research is being done to enhance the quality of the solution.

Acknowledgments

We would like to thank the voluntary test subjects for the time they spent testing the fall detection prototype. The feedback and test results are very valuable for the further development of the detection system. We would particularly like to thank Dr. med. Nicolette Fritsch-Wagner for her assistance during the validation of our ECG sensor and for medical advice on ECG analysis.

Paper partially funded by The Ministry of Economy and Competitiveness (Spain) and the FEDER Fund, under the National Program for Research, Development and Innovation, Societal Challenges Oriented, Project DArDOS TIN2015-65845-C3-3-R and Project FAME (RTI2018-093608-B-C33).

Appendix A
Datasets

[20] is a dataset repository which contains: fall simulation data, fall analysis, IoT-TEG generated test events and fall simulation video clip.

References

[1] World Health Organization (WHO), "Fact - sheets : Falls," http://www.who.int/news-room/fact-sheets/detail/falls, Last access: 08-07-2018.

[2] L. La Blunda and M. Wagner, "Fall-detection belt based on body area networks," in *Proceedings of Biotelemetry XXI (ISOB conference)*. Leuven (Belgium): Faculteit Ingenieurswetenschappen KU Leuven, 2016, vol. 1, pp. 21–24.

[3] ——, "The Usage of Body Area Networks for Fall-Detection," in *Proceedings of the eleventh International Network Conference (INC 2016)*. Frankfurt am Main (Germany): Alekseev, S. and Dowland, P. S. and Ghita, B. and Schneider, O., 2016, pp. 159–163.

[4] L. Gutiérrez-Madroñal, L. La Blunda, M. F. Wagner, and I. Medina-Bulo, "Test event generation for a fall-detection iot system," *IEEE Internet of Things Journal*, 2019, doi: 10.1109/JIOT.2019.2909434.

[5] EsperTech, "Espertech website," http://www.espertech.com/esper/index.php, Last access: 08-08-2018.

[6] L. Gutiérrez-Madroñal, I. Medina-Bulo, and J. J. Domínguez-Jiménez, "IoT–TEG: Test event generator system," *Journal of Systems and Software*, vol. 137, pp. 784–803, 2018.

[7] L. Gutiérrez-Madroñal, "Generación Automática de Casos en Procesamiento de Eventos con EPL - Authomatic Generation of Cases in Event Processing using EPL," Ph.D. dissertation, University of Cadiz, 2017.

[8] N. Leveson, *Engineering a safer world: Systems thinking applied to safety*. MIT press, 2011.

[9] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 1, p. 66, Jul, 2013.

[10] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, *Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information*. In Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on (pp. 138-143): IEEE, 2009.

[11] A. Collado-Villaverde, M. D. R-Moreno, D. F. Barrero, and D. Rodriguez, "Machine Learning Approach to Detect Falls on Elderly People Using Sound," in *Advances in Artificial Intelligence: From Theory to Practice*, S. Benferhat, K. Tabia, and M. Ali, Eds. Cham: Springer International Publishing, 2017, pp. 149–159.

[12] M. Lüder, G. Bieber, and R. Salomon, "Sturzerkennung mittels Luftdruck- und Beschleunigungssensorik Air Pressure- and Acceleration-Based Fall Detector: 2. Deutscher AAL-Kongress mit Ausstellung - Technologien, Anwendungen ; 27. - 28. January 2009 in Berlin; Tagungsbandbeiträge," 2009.

[13] H. Gjoreski, A. Rashkovska, S. Kozina, M. Lustrek, and M. Gams, "Telehealth using ECG sensor and accelerometer," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, pp. 270–274, 2014.

[14] S. Kozina, H. Gjoreski, M. Gams, and M. Luštrek, "Efficient Activity Recognition and Fall Detection Using Accelerometers," in *International Competition on Evaluating AAL Systems through Competitive Benchmarking*, vol. 386. Springer, pp. 13–23, 2013.

[15] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, vol. 14, no. 7, pp. 12 900–12 936, 2014.

[16] L. Gutiérrez-Madroñal, A. García-Domínguez, and I. Medina-Bulo, "Evolutionary mutation testing for IoT with recorded and generated events," *Software: Practice and Experience*, vol. 0, no. 0, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2629

[17] Y. Jia and M. Harman, "An analysis and survey of the development of mutation testing," *IEEE Transactions on Software Engineering*, vol. 37, no. 5, pp. 649–678, Sept, 2011.

[18] T. Jämsä, M. Kangas, I. Vikman, L. Nyberg, and R. Korpelainen, "Fall detection in the older people: From laboratory to real-life," *Proceedings of the Estonian Academy of Sciences*, vol. 63, no. 3, p. 253, 2014.

[19] L. La Blunda, "Die Sturzerkennung basierend auf Body Area Networks," Master's thesis, Frankfurt Univeristy of Applied Sciences - Germany, 2016.

[20] L. Gutiérrez-Madroñal, "Github repository," https://github.com/lorgut/FallDetectionData, 2018.

[21] N. Fritsch-Wagner, "Private communication - ECG validation and consultation in medical lab," 2017.

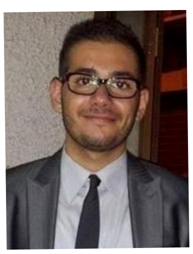
[22] R. Antonicelli, C. Ripa, A. M. Abbatecola, C. A. Capparuccia, L. Ferrara, and L. Spazzafumo, "Validation of the 3-lead tele-ECG versus the 12-lead tele-ECG and the conventional 12-lead ECG method in older people," *Journal of Telemedicine and Telecare*, vol. 18, no. 2, pp. 104–108, 2012.

[23] F. Sajid Butt, "Fall-detection using machine learning techniques on ECG-signals," Master's thesis, Frankfurt Univeristy of Applied Sciences - Germany, 2019.

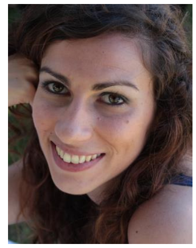
[24] U. Banu, G. Patil, and F. Ruksar, "A survey on sources of noise and advanced noise removal techniques of biosignals," vol. 7(2). International Journal on Emerging Technologies (Special Issue on ICRIET-2016), 2016, pp. 8–13.

[25] International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO), "Medical electrical equipment-part 1: General requirements for basic safety and essential performance," *IEC 60601-1-11, 2015*.

[26] V. Balgos, "A Systems Theoretic Application to Design for the Safety of Medical Diagnostic Devices," Master's thesis, University of Missouri, Columbia - USA, 2012.



Luigi La Blunda received his Bachelor of Engineering degree in Engineering Informatics in 2013 and his Master of Science degree in Barrier-free Systems, with the specialization in Intelligent Systems, in 2016 at the Frankfurt University of Applied Sciences (Germany). He has worked as a tutor in the laboratory experiments of electrical engineering, electronics, and in the lectures of software engineering analysis and design. In 2014 Luigi La Blunda became a research assistant in the WSN & IOT research group at the Frankfurt University of Applied Sciences (Germany). In addition, he supervises the project lecture Smart Sensor Network Systems in the master programs Barrier-free Systems - Intelligent Systems (BaSys), High Integrity Systems (HIS) and Computer Science. The aim of this course is to develop a running Wireless Smart Sensor Network (WSN) prototype. His research areas are Internet of Things and Wireless Sensor Networks, with specialization in fall analysis and detection. Since 2016 he is a PhD candidate in the PhD program Engineering Informatics at the University of Cádiz (Spain).



Lorena Gutiérrez-Madroñal received her first-class Honours Degree in Computer Systems Management in 2007, her BSc in Computer Science in 2009, her Master of Advanced Studies in Computer Science in 2010 and her PhD in 2017 at the University of Cádiz (Spain). She has been working at the Department of Computer Science and Engineering as a full time lecturer since 2009. Her research is focused on the Internet of Things and test event generation for any event-processing program. In order to prove the usability of the test generated events, she is using them to apply mutation testing to EPL query languages, such as the Event Processing Language (EPL). She has participated in research projects, all involved in software engineering related aspects. She has served in program and organizing committees at different conferences. She is a researcher of the UCASE Software Engineering Research Group.

Matthias F. Wagner received a Diploma and a Dr. rer.nat. in Physics from the Johannes Gutenberg - Universität Mainz (Germany). He was head of Measuring Technology Software Development at Hottinger Baldwin Messtechnik (HBM) in Darmstadt (Germany) from 1990 until 2002. 2002 he was appointed as Professor of Computer Science at the Frankfurt University of Applied Sciences in Frankfurt am Main (Germany). Since 2005 he is Program Director of the international M.Sc. program "High Integrity Systems". Since 2017 he is serving as Vice-Dean for Research and International Relations of the FB2, Department of Computer Science and Engineering. Since 2010 he is head of the Research Group Wireless Sensor Networks & Internet of Things (WSN & IoT). His research interests cover Safety Critical Computer Systems, Smart Sensor and Actuator Networks, Software and Systems Engineering and Computational Science and is supported by research stays at the UCASE Software Engineering Research Group of the Universidad de Cádiz (Spain) and the Dipartimento di Fisica e Astronomia of the Università degli Studi di Firenze (Italy).

Inmaculada Medina-Bulo received her PhD in Computer Science at the University of Seville (Spain). She has been an Associate Professor in the Department of Computer Science and Engineering of the University of Cádiz (Spain) since 1999. She has been a member of the Council of the School of Engineering (ESI) as well as a Socrates/Erasmus Program Coordinator for several years. From July 2010 to July 2011 she was appointed Degree Coordinator for the Computer Science Studies and a member of the Board of the ESI. Since September 2013 she holds the post of Chief Information Officer of the University. Her research was supported by research stays at the USA, the UK and Germany. She has served in program or organizing committees at different conferences and journals. She has published numerous papers in international journals, and international conference and workshop proceedings. She is the main researcher of the UCASE Software Engineering Research Group. Her main research interests are software verification, software testing, web service compositions, model-driven engineering and complex event processing. She has coordinated the development of several open source testing tools, such as the MuBPEL mutation testing tool for WS-BPEL, the GAmeraHOM tool for locating "hard-to-kill" mutants, the Rodan test case generation tool and the Takuan dynamic invariant generator for WS-BPEL. She has participated in and leaded research projects, all involved in software engineering related aspects.