A wearable fall-detection system based on Body Area Networks for smart cities

Luigi La Blunda[a], Lorena Gutiérrez-Madroñal[b], Matthias F. Wagner[a], Inmaculada Medina-Bulo[b]

[a] *WSN and IOT Research Group Frankfurt University of Applied Sciences, Nibelungenplatz 1, 60318 Frankfurt am Main, Germany*
[b] *UCASE Software Engineering Research group, University of Cadiz, Av. Universidad de Cadiz, 10, 11519 Puerto Real, Spain*

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 inhabitant which would significantly enrich the development of smart cities and smart factories where fall-events 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 STAMP (System-Theroetic Accident Model and Processes) will be applied.

Keywords: e-Health, fall-detection, Body Area Network, safety, STAMP, Smart City

Email addresses: `l.lablunda@fb2.fra-uas.de (corresponding author)` (Luigi La Blunda), `lorena.gutierrez@uca.es` (Lorena Gutiérrez-Madroñal), `mfwagner@fb2.fra-uas.de` (Matthias F. Wagner), `inmaculada.medina@uca.es` (Inmaculada Medina-Bulo)

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), 646,000 fatal falls are estimated to be the second leading cause of accidental or unintentional death worldwide each year. People over 65 suffer the most fatal falls. Another high risk group is children. Taking into consideration their evolving developmental stages, the increasing curiosity to explore the environment or the inadequate adult supervision lead to fall-events [**?**]. 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 from height while maintaining windmills. Mostly windmills are installed in uninhabited areas where no immediate rescue measures can be taken. Considering these events the consequences can be fatal for the affected people. The World Health Organization stated that annually 37.3 million fall-events are severe enough to require medical treatment [**?**]. Fall-events leads 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 of movement 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 was 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 time stamp and the sensor node of the ECG-harness provides the analog value of the voltage including time stamp. 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

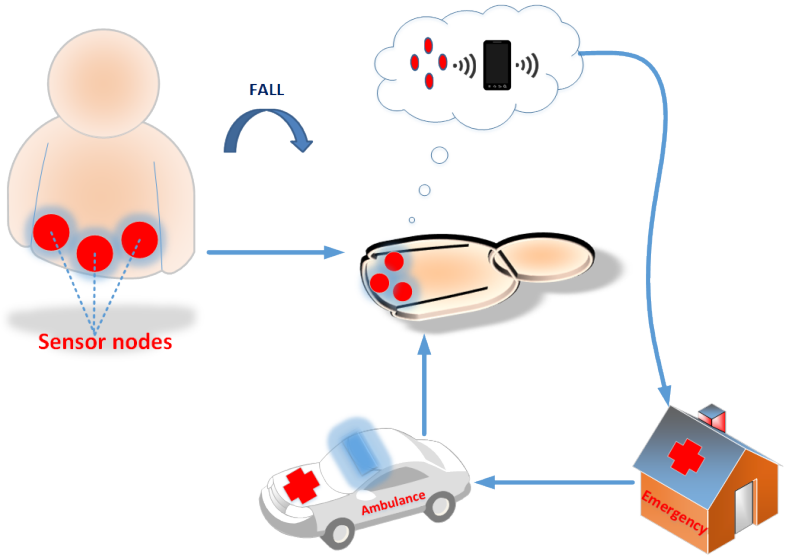

Figure 1: Escalation scheme simulation [**?** **?**]

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. In addition, EPLs are used to specify critical situations, extract information, and decide correctly on the basis of the collected data. Given that testing is essential for the system, test events are generated that simulate different types of falls. In literature, several types of falls are described and it is important to recognize all these falls. To generate the test events, the IoT-TEG tool is used which is capable to adapt the test events according to the type of fall. In terms of smart cities our expectancy is to significantly provide an important e-health service for smart cities which is an essential life-saver for the population. 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?

- Does the integration of our system as an e-health service in smart cities contribute to an improved quality of life for the population?
- How does the ECG-signal change during a fall situation? - Can relevant patterns be extracted that lead to a reliable fall detection?

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, Complex Event Processing (CEP) [**?**] and IoT-TEG tool [**? ?**]. 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 [**?**] which is applied on the fall-detection prototype. Finally, in section 6 a conclusion about this ongoing research 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. [**?**] illustrates 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 and this is not well accepted by many. 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. [**?**] contains wearable solutions worn on the body and are based on Body Area Networks (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. [**?**] illustrates wearable fall-detection systems using the sensor fusion method which comprises accelerometer and gyroscope data and built-in systems in form of smart phone sensors. For both categories of fall-detection solutions (context-aware & 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 into consideration the fact that it is an essential advantage to focus on wearable solution because of the system's functional independence the scientific work of Li et al. [**?**] will be explained subsequently. The solution of Li et al. [**?**] 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).

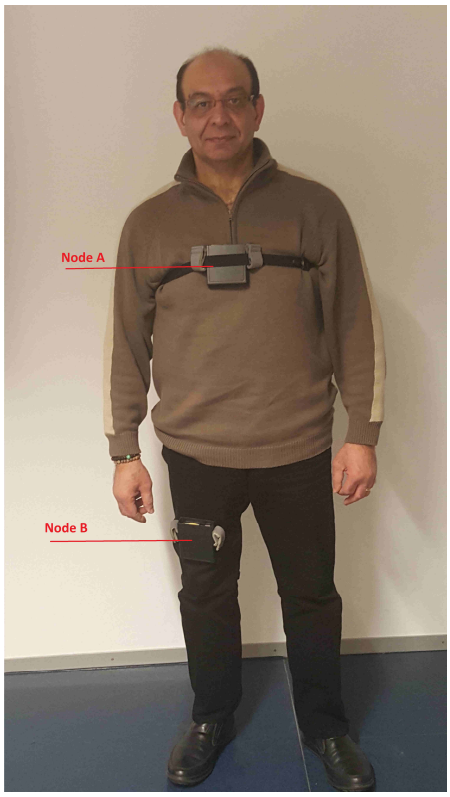

Figure 2: System architecture according to Li et al. [**?**]

Li et al. distinguish between two different 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. [**?**] to decrease the computational effort of the micro controller. 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 [**?**] uses a threshold based technique that is applied in the different phases of the algorithm. The weakness of this approach is to differentiate between jumping into bed and falling against wall with seated posture. Collado Villaverde et al. [**?**] 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. [**?**] which apply an air pressure sensor in addition to the accelerometer. The minor air pressure changes correspond to an altitude change of 25 cm which is a relevant indication in combination with accelerometer data of a possible fall-event. This sensor combination is incorporated in the hardware "StairMaster" of the Frauenhofer Institute. The hardware architecture proposed in [**?**] depicts a wearable solution which is worn on the hip and provides wireless data transmission via Bluetooth. Alternatively, the data can be stored on a Secure Digital (SD) card for evaluation purposes. Due to the fact that air pressure can be influenced by meteorological disturbances Lüder et al. [**?**] introduced an alternative architecture. This alternative approach incorporates an external barometric sensor as a reference, which is placed in a stationary altitude to compare both air pressure data (wearable & stationary). Another method for fall-detection based on sensor fusion techniques is described by Gjoreski et al. [**?**]. Accelerometer and electrocardiogram (ECG) data is used to detect fall-events. The solution in [**?**] is capable of identifying person's movements and fall-situations using wearable sensor nodes based on Bluetooth. These nodes are placed on the chest and thigh which is similar to Li et al's [**?**] approach. The

advantage of the proposed solution by Gjoreski et al. [**?**] 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. The results presented in [**?**] emphasize that the incorporation of the ECG-signal can significantly increase the system's reliability to detect fall-events. According to the analysis of Gjoreski et al. [**?**] differences in the ECG-signal in different postures were detected. Lower beat rates in the static positions lying and sitting compared to waling 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 [**?** **?**]. To detect a fall-event a typical physical behavior is used which comprises the following the phases (see Figure 3):

- Pre-fall-phase
- Falling phase
- Impact phase
- Post-fall phase

The pre-fall phase illustrates a stationary position of the person where the measured acceleration is around 1g ($9.81 m/s^2$). During the falling phase the acceleration decreases close to 0g ($0 m/s^2$), which depicts a similar behavior to a free fall. Upon the impact (impact phase), the acceleration reaches his maximum value for a short period of time. This signalizes that the person has suffered an impact. Subsequently, the acceleration is decreasing to around 1g ($9.81 m/s^2$) which represents the post-fall phase because after a fall the person is lying down. To apply this fall-detection pattern the accelerometer data of the sensor nodes

is used. Important to know is that for the impact detection the acceleration magnitude is used which is calculated with the following formula 1 and to determine the body's orientation the single axis values of the accelerometer (α_x, α_y, α_z) are analyzed:

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

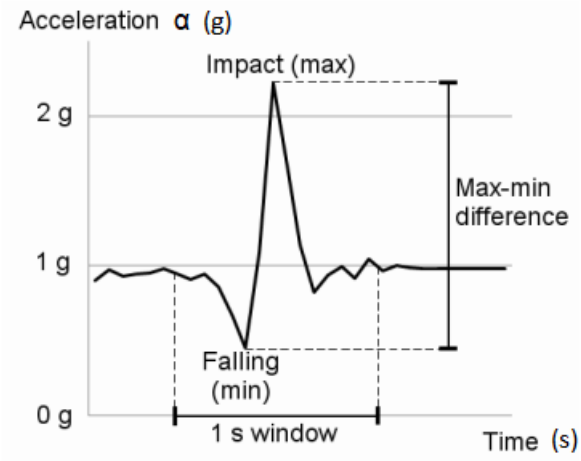

Figure 3: Acceleration during impact according to Kozina et al. [**?**]

Taking into consideration the body orientation the single acceleration axis (α_x, α_y, α_z) are essential to determine the person's body orientation. 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 this information with the acceleration magnitude (see Equation 1) the system is able to detect the fall and additionally determine the type of fall. For applying this knowledge the Event Processing Language (EPL) CEP of Espertech [**?**] was used which will be explained in the subsequent subsection 3.2.

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. The scientific papers by Li et al. [**?**] and Pannurat et al. [**?**] were chosen as a reference, 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 CEP

The EPL used in this ongoing research is CEP which is a stream-oriented language and executed by Espertech [**?**]. The main reasons for using CEP 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 worlstation and between 70.000 and 200.000 events per second on a notebook (according to Espertech) → This is an essential feature to simulate time-critical applications.

To define rules in CEP the incoming event should be defined, that means you need to specify incoming data for CEP-engine. After the event creation, rules should be defined to categorize the incoming input in fall-events or daily activity.

Example 1: Fall pattern based on Kozina et al. [**?**]

```
//1. Definition of the event - defines 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 & one lateral node) to apply the fall-detection, but in future this query will be extended for all the sensor nodes of the BAN. The four node BAN architecture is currently only used for redundancy purposes. In the example above 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 indicate that the person is in a stationary position in which the earth's gravity of 1g (9.81 m/s^2) acts on the body. Subsequently, it will be checked if the subtraction of the successive acceleration (a2.accelS1) and the first acceleration (a1.accelS1) within a time of 1 second is $>= <= 9.81$ m/s^2. If this condition is fulfilled, it is an indication that the person has suffered an impact to the ground. 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 nodes matches the EPL query and the values of the acceleration correspond to the same person.

3.3. IoT test event generator

IoT-TEG [**?** **?**] 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.

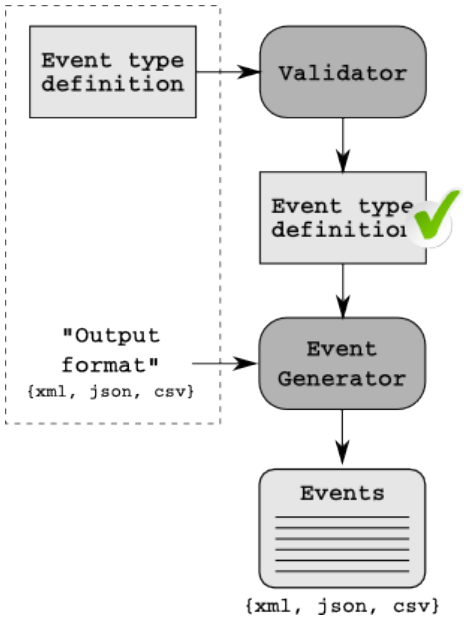

Figure 4: IoT-TEG Architecture [**?** **?**].

Previous studies suggested there were no differences in testing effectiveness between using events generated by IoT-TEG, or events recorded from various case studies [**?** **?**]. Moreover, thanks to its implementation, IoT-TEG can be used to do different type of test: functional, negative, integration, stress, etc; in deed an example of its usability can be found in [**?** **?**], where IoT-TEG has been used to apply the mutation testing [**?**]. 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:

1. Lack of data for testing,
2. needing specific values for the events, and
3. 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

```
<?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="yy-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="temperature" 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 `temperature`. These properties are defined as fields in the event type definition. The `created_at` field is complex type and `entry_id` and `temperature` are simple types. The property that is evaluated in the Example 2 queries is `temperature`.

Apart from the mentioned challenges that IoT-TEG solves in order to test event-processing programs, it incorporates a specific functionality for testing programs than use EPL of EsperTech. 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

The hardware architecture plays an important role in the development process to ensure the reliable functioning of the system. A first prototype was developed in this ongoing research which is introduced in [**?** **?**]. This paper presents an improved version of the prototype in [**?** **?**], which includes essential developments that contribute to reliable operation of the system. Additionally, the improved hardware design meets the requirement for patient compliance. This aspect should be taken into consideration to provide a user friendly design which facilitates the freedom of movement for the user. As a safety critical system for medical purposes, a redundant hardware design should be provided to protect the system against a total system failure which would have severe consequences for the user in case of fall-events. Referring to Jämsä et al. [**?**] 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).

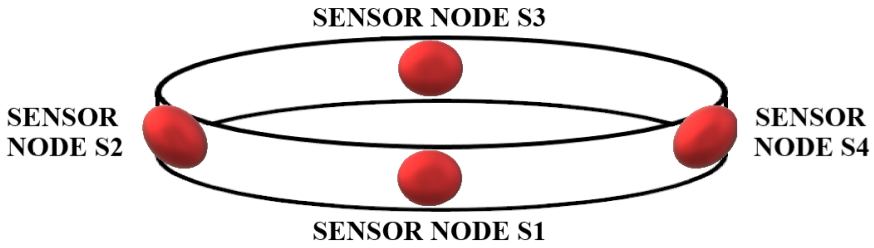

Figure 5: Four sensor nodes BAN - Belt [**?** **?**]

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 smart phone (central device). The dataset consists of the following information:

SensorID, X-Acceleration, Y-Acceleration, Z-Acceleration

- **SensorID** → is used to identify to which sensor node the incoming dataset belongs to.
- **X-Acceleration** → represents the acceleration value α_x in X-direction which includes the unit m/s^2.
- **Y-Acceleration** → represents the acceleration value α_y in Y-direction which includes the unit m/s^2.
- **Z-Acceleration** → represents the acceleration value α_z in Z-direction which includes the unit m/s^2.

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.

As an example, a person is taken who has suffered a frontal fall because of a fainting. Analyzing the fall-event, it can be determined that the fall caused by a fainting leads firstly to a left rotation of the person's body (pre-fall phase). This event triggers the fall of the person (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 Equation 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 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 [**?**]. Taking into consideration the test protocol stated in section 3 typical fall-events

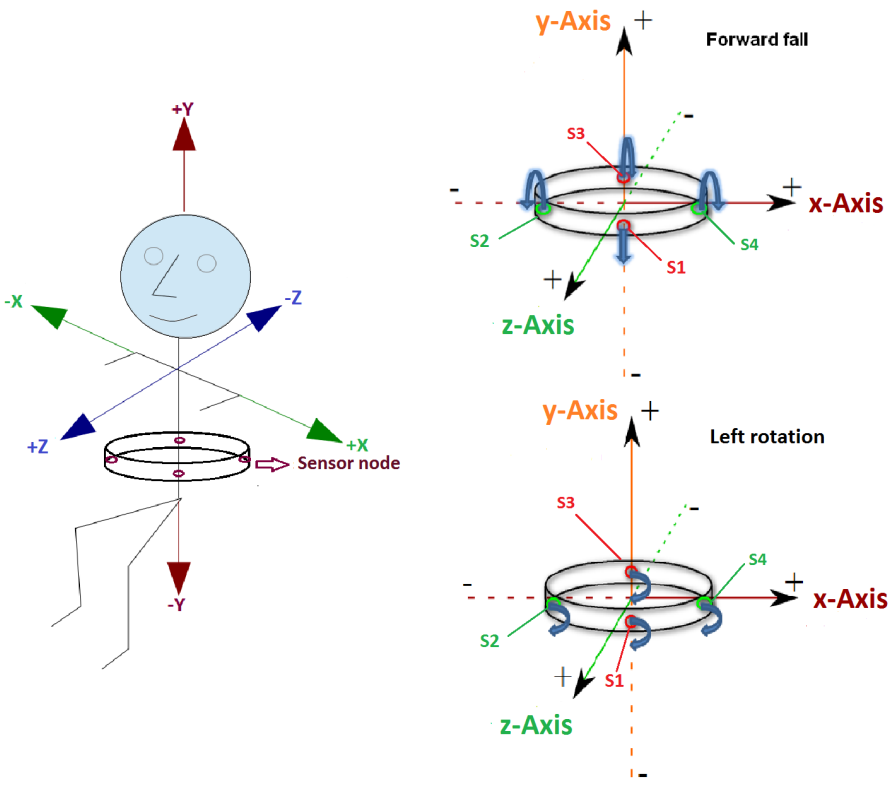

Figure 6: Three axis reference scheme [**?** **?**]

(e.g. rolling out of the bed) in nursing homes and hospitals can be detected by this solution.

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: *fall against wall* (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 fall test for a period of 2 minutes, the FAW fall type. The analyzed data and videos can be found in [**?**].

In this analysis the following steps have been done:

Study of the values

Given that the sensor 1 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 impacts of the falls, 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.

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 behaviour during FAW fall. The range of extracted values are a set of data that happen in less than a time window of 1 second, to meet the fall rule of [**?**] described in Equation (1), see Section 3. The Table 1 shows the acceleration value during one FAW fall of person 1 and person 2.

The first and fourth columns of the Table 1 represent the seconds and milliseconds ($s.ms$) when the acceleration was measured. The second and fifth columns show the acceleration values (m/s^2) and in the third and sixth columns are deployed the normalized acceleration values ($N(m/s^2)$) according to the fall maximum value. In Figure 7 a comparison of the normalized acceleration behavior during the previous FAW falls of person 1 and person 2 is shown. These comparisons show a similar behavior of the acceleration during the FAW fall.

Time ($s.ms$)	Accel. (m/s^2)	N. Accel. ($N(m/s^2)$)	Time ($s.ms$)	Accel. (m/s^2)	N. Accel. ($N(m/s^2)$)
57.311	3,63	0,01	25.254	8,53	0,09
57.359	3,43	0,00	25.303	10,79	0,19
57.408	7,85	0,17	25.352	11,28	0,22
57.506	10,99	0,30	25.401	13,83	0,34
57.506	6,47	0,12	25.449	9,42	0,13
57.554	4,61	0,05	25.503	8,63	0,09
57.603	4,22	0,03	25.546	0,89	0,10
57.651	22,86	0,77	25.596	8,04	0,07
57.700	28,84	1,00	25.645	13,93	0,35
57.749	21,78	0,72	25.693	22,7	0,76
57.800	9,03	0,22	25.742	10,59	0,19
57.846	10,79	0,29	25.791	8,63	0,09
57.900	10,79	0,29	25.841	8,34	0,08
57.944	9,32	0,23	25.888	6,67	0,00
57.996	13,15	0,38	25.937	17,46	0,51
58.042	20,99	0,69	25.986	27,76	1,00
58.186	28,25	0,98	26.082	11,58	0,23
58.188	27,47	0,95	26.084	11,38	0,22
58.240	10	0,26	26.134	10,5	0,18
58.286	10,99	0,30	26.181	8,83	0,10
58.334	11,67	0,33	26.230	12,75	0,29

Table 1: FAW fall acceleration, person 1 and 2

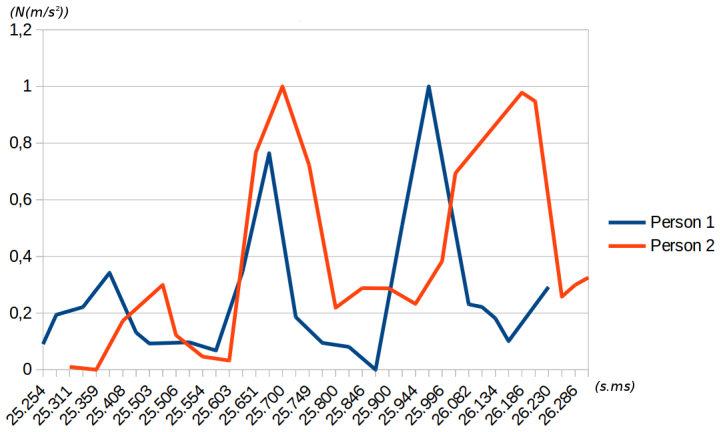

Figure 7: Acceleration comparison during FAW fall.

For the FAW, we have decided to define the acceleration behaviour with normalised 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].
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 decreases 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 normalised values has to be greater than 0.7.
5. Finally, the person is laying on the ground and the normalised 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 behaviour of the acceleration of all of them follows the same pattern.

To define the fall event

Once the fall acceleration behaviour has been observed, the next step is to define the fall event in order to generate test events with IoT-TEG [**? ?**]. 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 falls study. Given that the acceleration values follow

a specific behaviour, it is necessary to include the `custom_behaviour` property in the `<field>` element to define the behaviour of any event attribute; in this study, the acceleration. In the `custom_behaviour` property the path to the file that includes the behaviour of the event attribute has to be written. The Example 3 shows the complete fall event definition (FallEventType).

Example 3: Fall event type definition

```
<?xml version="1.0" encoding="UTF-8"?>
<event name="FallEventType">
<block name="feeds" repeat="100">
<field name="acceleration" quotes="false" type="Float"
custom_behaviour="/Path/To/Rule/File"></field>
</block>
</event>
```

IoT-TEG includes a new functionality, which has been implemented to simulate the desired behaviour of an event attribute with a `custom_behaviour` property in its event type definition. This functionality allows to generate values of the event attribute following a behaviour that the user has described in a file. In order to explain how the user has to define the desired behaviour of an event attribute, we are going to use the FAW fall behaviour 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 behaviour rules definition file, the number of events involved to simulate the behaviour is 20. In the FAW fall example, 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 behaviour rules. They can be defined in the file where the behaviour 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 behaviour during a FAW fall three variables are defined: `Base`, `ImpactWall` and `Impact`. Given that the acceleration behaviour during a FAW fall has being done according to the normalised 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, $1g \approx 9.81m/s^2$. That is the fixed value of the `Base` variable. To determine the values of the impacts, we have taken into account that the normalised 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 remained events, three events, for the sixth rule. We have to assign zero to the weight `weight="0"` to indicate how the remained 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. The `sequence` property values are `inc`, to increase the value, or `dec`, to decrease the value.

Example 4: Rules to define a FAW fall

```
<?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 behaviour rules that follow the normalised acceleration values can be defined. Given that we have considered to define the FAW fall behaviour rules according to the normalised 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 behaviour of the normalised 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.

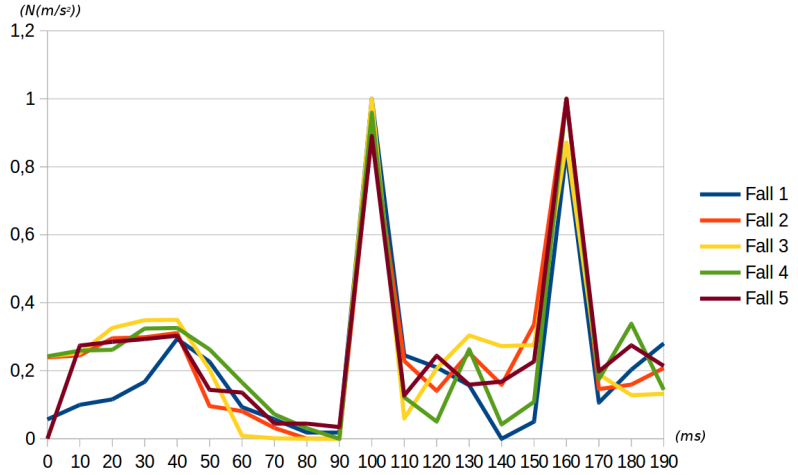

Figure 8: IoT-TEG generated FAW falls.

4.3. Sensor fusion

Analyzing the literature regarding fall detection, it was pointed out that the existing solutions show certain weaknesses in the reliable detection of falls [**? ? ? ? ? ?**]. Considering the approach of Gjoreski et al. [**?**], the results with the fusion of physical sensors and the ECG-sensor show a reliable fall detection. Especially the evaluation of the ECG-signal leads to an essential improvement of the system's accuracy. According to [**?**] 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 which provides an analog signal to the microcontroller. The successive illustration

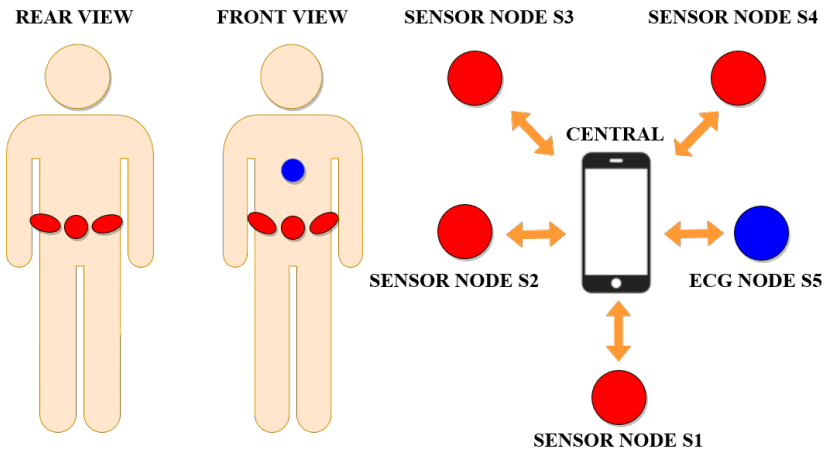

Figure 9: ECG - BAN

depicts the BAN structure of the update prototype architecture. 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 smart phone (central) via Bluetooth Low Energy. The sensor nodes S1 to S4 send the acceleration data and the ECG-sensor sends analog values (representation of ECG-signal) to the central device. Fusing these 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 own ECG-measurements must be compared with measurements taken by a medically certified clinical ECG-device, in order to guarantee the correctness of the ECG-values. Based on Gjoreski et al. [**?**] the ECG-signal is 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

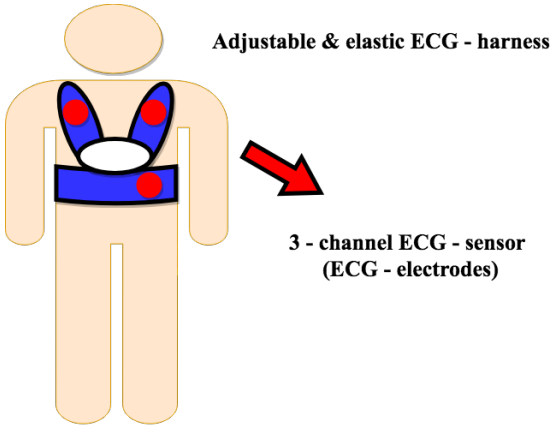

Figure 10: ECG - Harness

ECG-harness the measurements were repeated using the harness solution. A more stable signal resulted, especially during walking 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 [**?**]. According to a study by Antonicelli et al. [**?**], 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 [**?**] 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 [**?**] and [**?**] as part of the master thesis [**?**] in progress. The aim of this work is 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. A machine learning analysis is also planned in future.

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 prototype; problems related to its hardware and software. Talking about the software, the synchronization in the prototype is an issue; we have founded the following problems:

- In one hand, the falls from the beginning always should be discarded because the sensors are sending a lot of data. The Table 2 will be used to explain the problem. Let us say that in "fall 1", there are more than 30 values that define the fall, and later in "fall 3", there are 12 values that define the fall. So, in order to analyze the falls and compare the acceleration behavior, it is difficult to work with that information. A comparison of two FAW falls with the synchronization problem is shown in Figure 11; the sensor 1 acceleration values for the first FAW fall are colored in blue and the sensor 1 acceleration values for the second FAW fall are colored in red. The first fall is one fall from the beginning of the simulation, and the second one is from the middle of the simulation.

Time ($s.ms$)	Accel. (m/s^2)	N. Accel. ($N(m/s^2)$)	Time ($s.ms$)	Accel. (m/s^2)	N. Accel. ($N(m/s^2)$)
25.561	9,81	0,11	25.546	8,73	0,1
25.561	9,81	0,11	25.596	8,04	0,07
25.562	11,38	0,17	25.645	13,93	0,35
25.564	11,38	0,17	25.693	22,76	0,76
25.609	15,11	0,33	25.742	10,59	0,19
25.610	15,11	0,33	25.791	8,63	0,09
25.610	15,11	0,33	25.841	8,34	0,08
25.611	15,11	0,33	25.888	6,67	0
25.612	31,4	1	25.937	17,46	0,51
25.612	23,84	0,69	25.986	27,76	1
25.657	23,84	0,69	26.082	11,58	0,23
25.657	23,84	0,69	26.084	11,38	0,22
25.658	23,84	0,69			
25.659	23,84	0,69			
25.660	9,52	0,10			
25.660	9,52	0,10			
25.706	9,52	0,10			
25.706	9,52	0,10			
25.707	9,52	0,10			
25.707	9,52	0,10			
25.708	23,74	0,69			
25.709	23,74	0,69			
25.756	23,74	0,69			
25.756	23,74	0,69			
25.757	27,86	0,85			
25.758	27,86	0,85			
25.758	9,52	0,1			
25.759	9,52	0,1			
25.760	9,52	0,1			
25.804	7,06	0			
25.805	7,06	0			
25.805	7,06	0			
25.806	7,06	0			

Table 2: Synchronisation problem, fall 1 and fall 2

The acceleration values show that in less than 250 milliseconds there are more than 30 values from “fall 1”, and in more than 350 milliseconds there are 12 values from “fall 2”. There is a lack of synchronization not only in

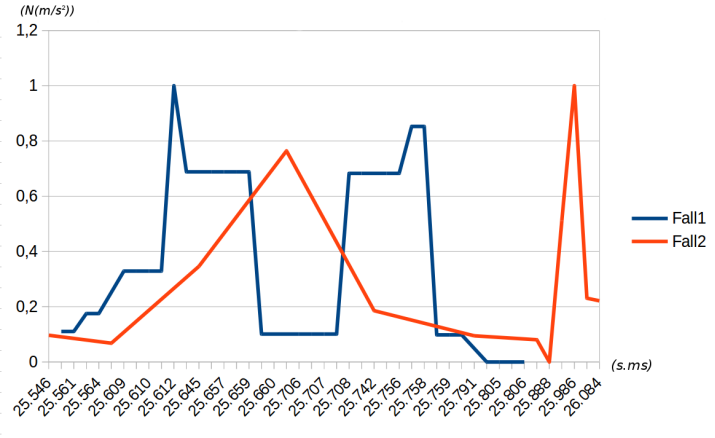

Figure 11: Synchronization problem; comparison of acceleration values during two FAW falls

the amount of data, but also in the time.

- On the other hand, some sensors transmit more data than the others; no matter the fall, the sensors are not sending the same amount of data, even sometimes there is no data. The four sensors were working while the FAW fall-simulation, but the obtained acceleration values were from three of them, one of the sensor does not transmit data in one moment of the simulation, see Figure 12.

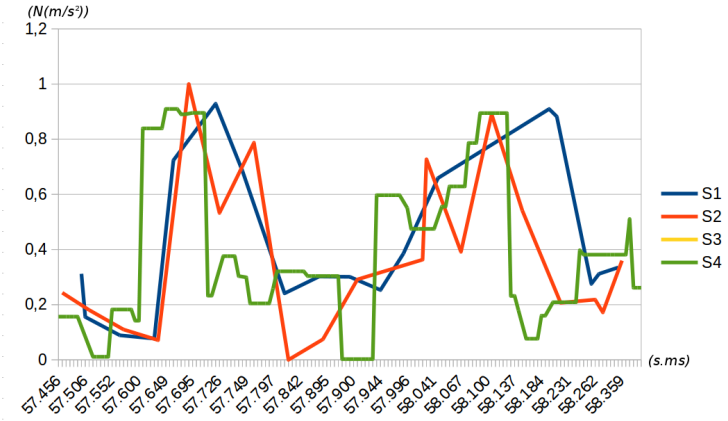

Figure 12: Synchronization problem; acceleration values from the four sensors during a FAW fall

Concerning the hardware of the prototype, there are some aspects that need to be improved. One point 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 are thinking about elderly people who live alone or employees who work in hazardous working conditions, we have to ensure a permanent monitoring of the parameters in order to provide quick intervention in life-threatening situations.

If we consider the used ECG-sensor architecture, we encountered differences in signal quality that includes noise and baseline shifting (see Figure 13). The reasons for these distinctions in signal quality were explained by the supervising physician. Baseline wander (BW) disturbances are caused by variations in

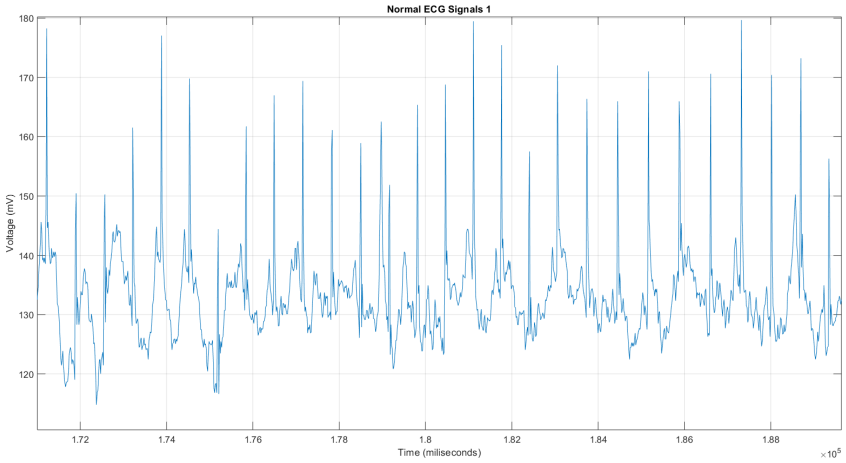

Figure 13: Baseline shifting ECG-signal

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. The certified medical devices have an integrated filter which can be applied to smooth the signal [**?** **?**]. To solve this problem the application of a Low & High pass filter in our setup is considered.

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 order to ensure a better simulation, the test subjects will lie on the ground for at least 2 seconds after simulating a fall. In a real situation, if a person falls and is able to stands 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 2 seconds in our test scenarios is helpful to perform an accurate fall analysis.

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) [**?**]. 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. This technique should be applied already in the early design phases to counteract developmental errors [**?**]. In the following STAMP will be applied in some architectural parts of the fall-detection system for functional safety validation.

STAMP (System-Theoretic Accident Model and Processes) 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 [**?**] 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
- Erros 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. The basic principle of the control structure is depicted in the successive illustration (see Figure 14).

The graphic shows that the control structure consists of the following components:

- Controller
- Actuators
- Controlled Process
- Sensors

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

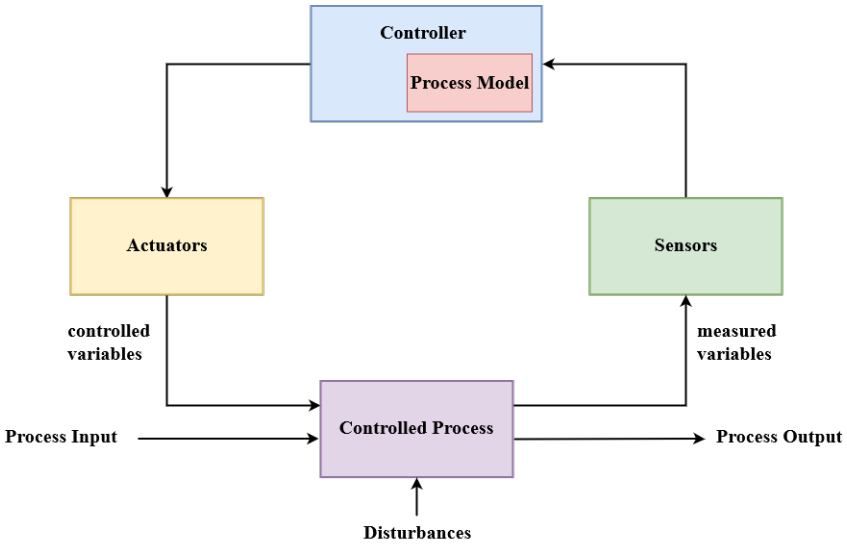

Figure 14: STAMP - Basic control loop structure according to Leveson [**?**])

process state. This information is used by the master to initiate a control action (controlled variables) to run the system within predefined limits.

To apply STAMP the following steps must be followed:

- **Definition of accidents:** Person has fallen and the fall was not detected by the system.
- **Definition of hazards:** Data from 1 to 5 sensors of the belt are not sufficiently correlated in time (synchronization error).
- **Definition of safety requirements & 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. 15) depicts the control loop of the controlled process, the movement of the person. The sensors provide sensor data to the controller (smartphone). 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.

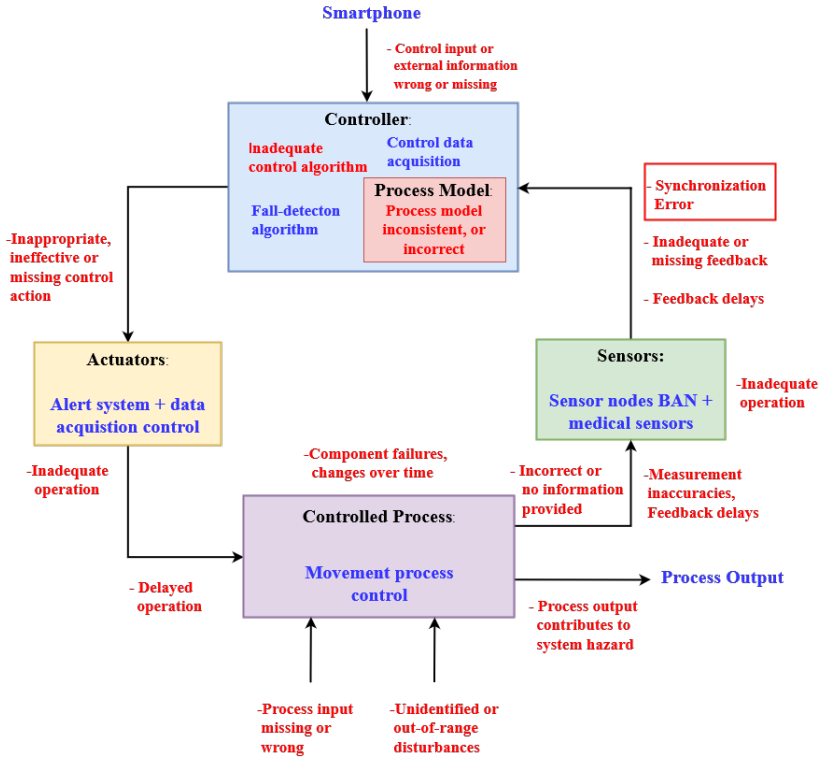

Figure 15: System Safety Control Structure - Hierarchy level 1) → Black label (system components), blue label (description of system components), red label (possible hazards)

The lower level control loops (see Figures 16 and 17) 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.

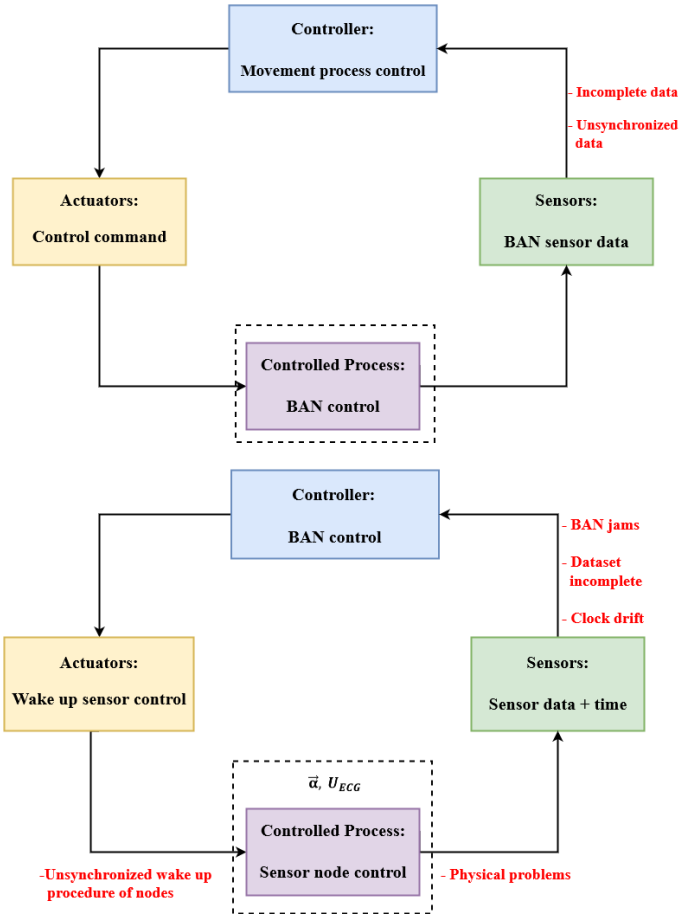

Figure 16: Safety Control Structure - Hierarchy level 2 & 3

To provide a more detailed hazard analysis two more control loop layers were created which are illustrated in the successive illustration (see Figure 17). The fourth level of the system's control structure zooms into Sensor node control (controlled process in Figure 16) which is the control unit (controller). Considering this control layer the controller receives the sensor's services and characteristics which are provided via Bluetooth Low Energy (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).

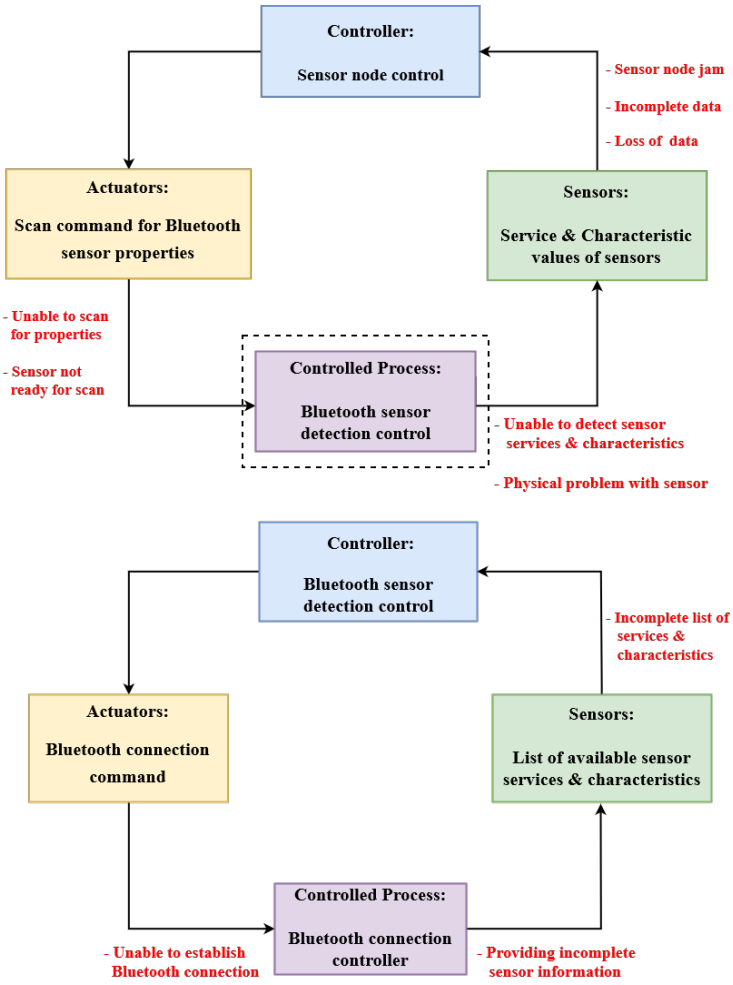

Figure 17: Safety Control Structure - Hierarchy level 4 & 5

Taking into consideration the fifth level of the control loop which is depicted in Figure 17 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 15, 16, 17) 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 → 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 → 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 → It must be ensured that the sensor information (services & characteristics) is transmitted completely.
- Fifth level control structure → 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 15) and lead to violation of the safety constraint Δt <= 50ms 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 STAMP, which represents a fraction of the system. We will use it for all parts of the complete architecture.

6. Conclusion

This work presents a wearable solution for fall detection that enhances the benefits of smart cities in terms of e-health. The growing population and the urban lifestyle require a smart health care system which is capable to provide a fast and efficient medical care. Thinking about the benefits that e-health brings to smart cities, it becomes clear that minimal response times and rapid intervention in emergency situations can significantly improve the population's lives. A typical use case is are smart factories that uses fully automated processes that cause the deployment of a small number of personnel and therefore the workers usually have to work alone in hazardous working conditions. In the case of a fall-event during the night shift, the integration of the portable fall-detection system (e-health) into the smart city infrastructure can provide immediate intervention.

The tests performed with the test persons provided positive feedback regarding the use of this solution and a high acceptance of the system in application with smart cities was achieved. These results also show that falls are a general problem in our daily life, which are underestimated and can lead to serious injuries or even fatal consequences.

The test measurements carried out with the additional ECG-sensor were positive and confirmed the approach of Gjoresk et al. [**?**] which allows a more accurate fall detection and, if necessary, a fall prediction. Additionally, the usage of the IoT-TEG tool [**? ?**] facilitates the generation of events to recognize falls which are based on [**?**]. 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 [**? ?**] has the ability to adapt the behavior of the analyzed event attribute, because it was designed to test any type of

system using test events. In the further development process it is planned to use machine learning and complementary Complex Event Processing (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 [**? ?**] is in constant development to ensure an accurate analysis of falls, further functionalities will be integrated in the future to meet all test requirements. In order to improve the tool, it is necessary to analyze further fall events.

However, first the synchronization problem of the BAN must be solved, which can compromise the system's functionality. In order to solve this problem, a new hardware platform should 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 the requirements of patient compliance (see subsection 4.1).

Acknowledgment

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 explaining how important the use of medical sensors (ECG sensors) can be for the detection of falls.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Dataset

References

mybibfile