Simulation of IoT-Oriented Fall Detection Systems Architectures for In-Home Patients

Renato F. Bulcão-Neto, Paulo G. Teixeira, Bruno G. A. Lebtag, Valdemar V. G. Neto, Alessandra A. Macedo and Bernard P. Zeigler

***Abstract*—Fall detection (FD) systems enable rapid detection and intervention for people who experience falls, a leading threat to the elderly's health and autonomy. Most of these systems conform to an IoT reference architecture which may include multiple sensing mechanisms to balance the advantages and drawbacks of each alternative. However, developing such a heterogeneous system may be costly and quite resource and time-demanding. This paper presents a Discrete Event System Specification (DEVS) simulation model for FD systems that compares the accuracy of nine different systems architectures that combine traditional wearable and non-wearable sensing devices in the acquisition layer. We perform simulations for each architectural arrangement using four public datasets of FD systems, totaling 36 simulations. Results reveal that an FD accuracy of 96.67% is possible with an investment of almost $6,000 US. Besides, spending 36 times less (around $150 US), designers and clients could acquire an FD system composed of wearable and non-wearable devices with an accuracy of 91%, i.e., only 5% less than the most expensive alternative.**

***Index Terms*—Fall detection, System architecture, Discrete event simulation, Internet of Things, Experimentation.**

I. Introduction

Due to the significant threat to the health and independence of adults aged 65 years and older, there have been many fall detection (FD) systems for preventing and detecting falls in later life [1], [2]. FD systems often capture data from sensing devices with accelerometers. Here, a person monitored must carry a wearable device that analyzes the acceleration of two or more axes and identifies sudden changes in acceleration as falls. As an alternative to wearable devices, Doppler radar-based FD systems [3], [4] rely on a non-wearable radar device that emits sound waves and identifies the acceleration of a person's body; again, a likely fall takes place when an abrupt variation in acceleration exists.

Regardless of which and how people's data is captured and interpreted as a fall, most FD systems implement an Internet of Things (IoT) reference architecture [5]–[7], which includes four layers: data acquisition, modeling, reasoning, and dissemination. Consider the situation in which Ilka, a 70-year-old frail woman, lives alone in a small house. Built upon multiple sensors, the FD system automatically monitors (acquisition and modeling), analyzes Ilka's activity patterns, and identifies events that might indicate a fall (reasoning). If Ilka suddenly falls, an appropriate alert is immediately sent to an application running on her daughters' smartphones (dissemination). Therefore, FD systems architectures based on IoT principles cover the whole life cycle of Ilka's activities data, i.e., from acquisition to dissemination.

However, usability issues may occur in the case of FD systems whose sensing mechanisms are primarily composed of accelerometers [8]. For example, consider that Ilka enters the bathroom carrying a device tight to her body (usually chest, waist, or wrist). Although it is often a low-cost device, the apparatus may cause discomfort, and there is no guarantee that Ilka does not take it off. On the other hand, despite its high cost, sensing based on Doppler radar is less obtrusive from the user's point of view and more effective in FD [4].

In this context, developing an IoT-oriented FD system with multiple sensing mechanisms can take advantage of a sensor's benefits (e.g., low cost) and overcome its drawbacks (e.g., obtrusiveness). On the other hand, this heterogeneity poses high demands when developing the data acquisition process (e.g., costly deployment and sensors maintenance), and there is no clue concerning the accuracy of FD results.

For those reasons, simulation models may help anticipate the performance of different FD systems architectures still in the system design phase [9] and reduce risks, time, and experiment costs. Unfortunately, there is little research on simulations of FD systems [10]–[14], despite the recent and increasing interest in such simulations.

This paper presents a Discrete Event System Specification (DEVS) simulation model [15], [16] for FD systems. Our DEVS model compares the accuracy of nine IoT-based FD systems architectures combining traditional wearable and non-wearable sensing devices in the acquisition layer. Using four public datasets for FD systems, we executed 36 different simulations. Results reveal that, among six out of nine assessed architectures, we can achieve more than 90% of FD accuracy with costs ranging from 150 to 6,000 US dollars.

In brief, our main contribution is a simulation model based on the DEVS formalism both for the specification and properties prediction (e.g., accuracy and cost) of IoT-oriented FD systems architectures at design time.

The organization of this paper is as follows. Section 2 presents related work and our contributions. Section 3 introduces background information. Section 4 details the IoT-based FD system architectures used in the simulations. Section 5 describes the simulation protocol. Section 6 discusses the findings, limitations, and threats to the validity of this study. Finally, Section 7 includes the final remarks.

Renato F. Bulcão-Neto, Paulo G. Teixeira, Bruno G. A. Lebtag and Valdemar V. G. Neto are with Universidade Federal de Goiás, Brazil. e-mail:rbulcao@ufg.br, valdemarneto@ufg.br.

Alessandra A. Macedo is with Universidade de São Paulo, Brazil. e-mail:ale.alaniz@usp.br.

Bernard P. Zeigler is with RTSync Corp. e-mail:zeigler@rtsync.com.

II. Related Work

Gutiérrez-Madroñal et al. [11] analyze the acceleration behavior in two types of falls based on a belt prototype worn on the patient's hip. They model fall-related event patterns and simulate these through an Event Processing Language (EPL). The simulation of behaviors to generate fall-related test events enhances the IoT-Test Event Generator (IoT-TEG) system. Comparatively, our approach can make use of the IoT-TEG's dataset if fall events are explicitly annotated. Otherwise, that dataset would be worthless because we do not reason over the raw data. The authors also agree with us regarding a more complex acquisition layer, i.e., with different types of sensors for fall simulation purposes.

To model simulate how different human body parts behave during a fall, Mastorakis and others [12] compare a fall velocity profile with a myoskeletal simulation model. Authors claim this approach is an efficient and customizable alternative for FD compared with strategies including dissembled fall datasets and low variability of people. Although our work on discrete event simulation differs far from the physics simulation approach, we agree that fall-related velocity/acceleration profiles can also be extracted from a more diverse acquisition layer, instead of solely from depth videos.

Noury & Hadidi [10] collected experimental data from a smart home for the elderly and modeled the behavior of data series using Hidden Markov Models. They exploited these models to produce artificial data series, but very similar to real data and in a flexible manner for addressing falls. Similarly, they also consider simulation as an essential tool in response to the costly and demanding conditions of field experiments, although not being exclusively focused on FD, but so on elderly daily activities in general.

A system developed by Makhlouf et al. [13] detects patients' heart rates and fall events. It performs a cause and effect analysis involving heart problems and falls, and also detects the patient's location. Differently of our simulation approach, the authors build a Petri net-based simulation model to validate their system with random and real data, with satisfactory results. However, as the system relies on wearable sensors data, the authors point out the need for improvements in the FD service, including different human positions and activities.

The novelty of our approach relies precisely on adopting DEVS simulation models for simulating a specific type of IoT-based system (FD systems). The differences between this type of IoT-based system and others rely on (i) how they are combined to make up the whole system, (ii) where they are deployed, (iii) and how the measured overall accuracy in the simulation directly impacts the final result. Given this is a critical domain (since failures could lead to injuries and even deaths of the monitored people), our study is a contribution because it proposes the adoption of simulation models for trading off the parameters mentioned above. Moreover, according to [17], [18], the main activities supported by FD systems are fall prediction, prevention, and detection. Our model contributes to detection activities since we can anticipate the overall accuracy delivered by different arrangements so that a professional or client can select the one with the highest accuracy in FD.

III. Background

This section introduces background information on the discrete event simulation paradigm and a reference software architecture for IoT systems development.

A. Discrete Event Simulation

A simulation model is a formal representation resulting from the observation of a real subject of interest (e.g., a system). Simulation models may be used for conducting experimental studies in Software Engineering, called simulation-based studies (SBS) [19]. SBS may anticipate the effects of a system's implementation when essential factors (e.g., risks, time, and costs) must be considered in the corresponding real implementation. As they usually run in virtual settings, SBS may also allow us to mitigate risks, time, and costs of experiments [9].

During the design and implementation of an SBS, multiple techniques may be adopted to model the behavior of a given system, such as discrete event simulation techniques. A discrete event simulation (DES) model describes the operation of a system as a succession of events that cause the system to change its state at discrete instants of time [15], [16]. For example, a new event is triggered whenever a simulation element generates output.

One of the most popular formalisms for modeling complex dynamic systems through discrete event abstractions is the Discrete Event System Specification (DEVS) [15], [20]. At this abstraction level, the next state of a system is defined based both on the event and the previous state of the system. Thus, DEVS-based models only consider states at which events occur, skipping over all intermediate points in time. One aspect that differentiates DEVS from other modeling formalisms is its approach to generically representing the total state of a system that varies in time as a function of the output values and state transitions of a model.

DEVS-based simulation models may be composed of two primary constructors: atomic and coupled models. The former can represent minor parts of a complex system, such as a sensor or even a small system, while the latter consists of combining different, communicating DEVS atomic models to create a more complex structure. For instance, if we want to simulate a human body, the organism is a coupled model composed of other related models (digestive system, respiratory system, etc.) that finally comprise atomic models, such as the stomach or lung. Then, coupled models support composability, i.e., the possibility of iteratively composing other atomic and coupled models to create complex structures. Coupled models are expressed as System Entity Structure (SES), enabling them to represent a family of different and reusable simulation models. In a pruning process, the user can select specializations in an SES which eventually results in an executable hierarchical coupled DEVS model, as shown later in Figure 1. SES files materialize the coupled models and specify how the inner systems (atomic or other coupled models) communicate among themselves [15].

B. IoT Reference Architecture

The core concept of IoT systems development is context information, which is any piece of information that characterizes the situation of a relevant entity in a user-application interaction [5]. It can be of several types (e.g., location and identity) and describe the situation of multiple types of entities (e.g., users). Moreover, it can be static, sensed, or derived from other simpler ones. These and other characteristics heavily influence the design aspects of IoT systems architectures.

The literature has synthesized various IoT systems implementations into the following layered reference system architecture [5]–[7]:

- **Acquisition:** this layer addresses how and where to get context information that will be consumed by an IoT application/system. Software components in this layer should implement techniques dealing with multiple particularities of the acquisition process including the acquisition's responsibility (pull/push), the frequency of data acquisition (instant/interval), the data source (sensor, middleware, or server), and the sensor type (physical/logical). *Depending on the combination of such techniques, the implementation of that complex scenario may be costly and effort- and time-demanding regarding sensor programming, deployment, and maintenance, for instance.*
- **Modeling:** it concerns the way the acquired context information is represented through modeling techniques. The choice of a specific modeling technique relies on particularities of the previously acquired data such as its diversity, freshness, imperfection, and the need for relationships and dependencies representation. *If a new sensing mechanism is introduced into an IoT system (e.g., video processing-based), this may require updates in information modeling and even the replacement of the modeling technique.* Popular context information modeling techniques (e.g., attribute-value pair and object orientation) are surveyed elsewhere [5], [21].
- **Reasoning:** it employs methods of deducing new knowledge based on the available modeled data. Similar to the modeling layer, the requirements of reasoning also emerge from characteristics of acquired data: imperfection and uncertainty. *A preprocessing step in the reasoning process may be necessary to deal with inaccurate collected data or missing values due to sensing inefficiencies.* This phase may also help improve reasoning performance in terms of efficiency, soundness, and completeness. Additional information about reasoning methods (e.g., first-order logic) is found in [5], [21].
- **Dissemination:** the delivery of context information to consumers is the main goal of this layer. *As the information to be delivered can be deduced or not, the same factors discussed in context acquisition are applicable in the development of context dissemination methods.* Consider the dissemination's responsibility and frequency, which can be implemented employing a consumer-initiated query, or an event-based publish/subscribe mechanism.

This reference architecture serves as guidance for orchestrating FD systems architectures, as we describe next.

TABLE I
ACCURACY AND PURCHASE PRICE OF THE SELECTED DEVICES.

Device	Description	Accuracy	Price
W1	Arduino UNO with ADXL335	92.7%	$33.99
W2	Two Samsung Galaxy Mini phones	95.29%	$178
W3	LG G Watch R / Samsung Galaxy S3	68%	$238
NW1	Two Microsoft LifeCam Cinema	95.1%	$275.96
NW2	Microsoft Kinect	79.91%	$114.95
NW3	SDR KIT 2500B	97%	$5,495

IV. IoT-based Architectures for Fall Detection of In-home Patients

A. The Architectural Arrangements

Recent studies reveal that the most common wearable IoT devices used in FD systems [22]–[27] are accelerometers (W1), smartphones (W2), and the combined usage of smartphones with smartwatches (W3). In parallel, the most common IoT non-wearable devices in FD systems [26], [28], [29] are conventional cameras (NW1), depth cameras (e.g., Kinect - NW2) [30], and Doppler radar (NW3) [31].

In our home scenario, an elderly person is monitored by one wearable device at a time (W1 to W3). Conversely, when the person is in the bathroom, one of these non-wearable devices (NW1 to NW3) can monitor her/his actions to instantaneously detect an eventual fall. This is a realistic scenario, since approximately 9% of elderly falls at home take place in the bathroom [32] and most of those falls are not detected (e.g., often people take off the wearable devices to shower.)

The accuracy of every wearable and non-wearable device present in the home scenario was collected from the literature in FD systems. We searched for the acquisition price of each device on the Amazon website on the same day so that this information could not impact the results of our experiment. The NW3 was the only device whose price had to be obtained through its manufacturer [31]. Table I maps each wearable and non-wearable device to its corresponding accuracy rate and acquisition price. Four highlights are noteworthy:

- the W1 device is an accelerometer-based equipment wirelessly connected to an Arduino board, as in [27];
- two smartphones represent the W2 device because a person should wear each one on right-side and left-side pockets, as we found in [33];
- the W3 solution is a smartphone wirelessly connected to a smartwatch a user wears, and FD may be based on the sensors data of any of these wearable devices [23];
- and two depth cameras (NW1) because of the signal obstruction for FD purposes found in [26].

There are advantages and drawbacks when exclusively equipping a home environment with wearable or non-wearable devices. Usability, cost, range precision, and signal obtrusiveness should be considered in this situation. For this reason, we combine W1 to W3 with NW1 to NW3, resulting in nine different architectural arrangements (ARCH) for FD systems presented in Table II. These arrangements exercise compositions of devices present in the literature.

TABLE II
FD SYSTEM ARCHITECTURES, THE COMBINATION OF WEARABLE AND NON-WEARABLE DEVICES, AND THE ARCHITECTURAL COST.

ID	W1	W2	W3	NW1	NW2	NW3	Cost
ARCH**1**	x					x	$5,528.99
ARCH**2**	x			x			$309.95
ARCH**3**	x				x		$148.94
ARCH**4**		x				x	$5,673.00
ARCH**5**		x		x			$453.96
ARCH**6**		x			x		$292.95
ARCH**7**			x			x	$5,733.00
ARCH**8**			x	x			$513.96
ARCH**9**			x		x		$352.95

In Table II, ARCH**1** to ARCH**3** combine one accelerometer attached to the user with each one of the non-wearable devices available for fall detection in the bathroom, i.e., cameras (NW1), one Kinect (NW2), and one Doppler radar kit (NW3). In turn, ARCH**4** to ARCH**6** suggest a user makes use of a smartphone equipped with internal sensors useful for FD detection, in combination with each non-wearable device located in the bathroom. Finally, the acquisition layer of ARCH**7** to ARCH**9** associates non-wearable devices with a smartphone and a smartwatch.

From these nine architectural proposals, we assess each one of the different combinations against given parameters, such as *acquisition cost* and *accuracy*. The next section describes how we model those architectural arrangements using DEVS.

B. Simulation Model for FD System Architectures

Methodological infrastructure setting. The FD system architectures mentioned above were modeled using a DEVS variant called DEVS Natural Language (DEVSNL), which runs on the MS4 Me modeling and simulation integrated development environment (MS4 Me IDE).

A formal structure called the System Entity Structure (SES) is governed by a small number of axioms that provide clarity and rigour to the simulation model [15], [34], [35]. SES supports hierarchical and modular compositions, allowing large complex structures to be built in a stepwise fashion from smaller, simpler ones. SES can also represent specialization and decomposition relationships and message flows among systems entities.

Representation of an FD IoT-based system using MS4 Me. Figure 1 illustrates a general IoT-based FD system architecture as we model the capabilities of SES. Central to this research, the acquisition layer comprises three main entities:

- *Human Body*: it reads input data from the simulation model and sends them to a WD or NWD, depending on the sensing device used in the simulation; and
- *Non-Wearable device (NWD) and Wearable device (WD)*: both capture the human body data and transmits it to the modeling layer.

The modeling layer includes a feature extractor entity, which fuses acquired data and forwards it to the reasoning layer. This, in turn, comprises a data evaluation entity, which decides whether a fall occurred or not and sends that data interpretation to the alert system of the dissemination layer.

If a fall is detected, the alert system notifies the caretaker entity and the emergency subsystem. Finally, the assistance center of the emergency subsystem receives a fall alert from the dissemination layer and forwards it as a message asking for an ambulance.

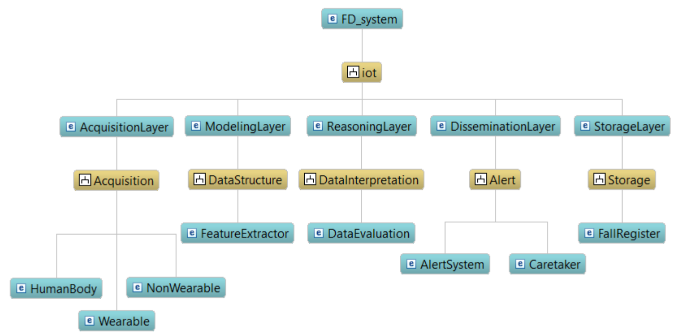


Fig. 1. SES tree representation for the FD system.

Once the SES of the FD system architecture is defined, the next step is the specification of the atomic models for each SES entity described in Fig. 1. Due to a large number of atomic models in our simulation model, we chose composing the main flow from a model of human body data acquisition and going to the fall detection stage. We describe them next. For space restrictions, we show in an external link [36] how atomic and coupled models are obtained from the SES structure.

Human Body. This is the FD system starting point. The human body atomic model reads one dataset at a time and implements the decision rule about which type of sensing device is handled as a data source (wearable or non-wearable). At the end of the process, it sends the data from the dataset to the specific sensing device.

Fig. 2 depicts the state transition structure of the human body atomic model as expressed in MS4 Me using DEVSNL and java tag blocks. The system starts reading the dataset file (state S0). While the end of the file is not reached, the system assigns data to the wearable device at the state S1, or to the non-wearable device at the state S2. After, the system returns to S0 to read the following data.

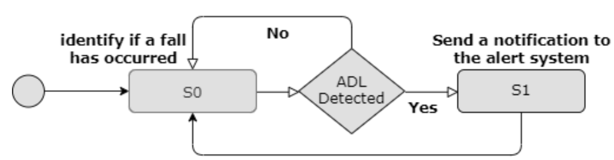


Fig. 2. State machine of the human body atomic model.

Data Evaluation. this atomic model waits for a message from the modeling layer (S0). Upon the reception of a message, it identifies whether a fall has occurred or not. Once a fall is detected, the status of the person being monitored is changed, and a message is sent to the dissemination layer (S1). If the system interprets a non-fall, it returns to a passive state and waits for a new message from the modeling layer. Fig. 3 illustrates the state transition structure of this atomic model.

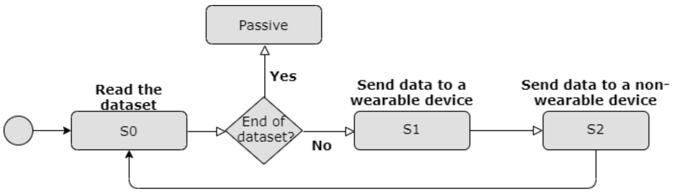


Fig. 3. State machine of the data evaluation atomic model.

For each piece of data obtained from the dataset representing an action (a fall or an activity of daily living – ADL), as in Fig. 2 and 3, a first verification is done to assess whether the action is performed inside the bathroom or not.

Next, we report the simulation protocol concerning planning, experiment execution, and results analysis.

V. Evaluation

In this section, we evaluate the nine architecture alternatives conceived for this study. The methodological procedures used in the experiment conforms to the guidelines defined by [37], such as follows: the experiment protocol planning, the collection and analysis of evidence, and the results discussion.

Supplemental material about the full protocol, each dataset, and the modeling of the FD system architectures used in the experimentation is available at https://bit.ly/30ynLo3.

A. Planning: Experiment Protocol

Scenario Description. The simulation takes place in the context of a home environment where older adults live alone. The scope of the simulation is the acquisition layer of an IoT-based FD system. These data are sent to other layers; however, these are abstracted. We simulate the monitoring using a combination of six devices, as shown in Table I.

An older adult may perform activities in two regions of her/his house. Among these actions, falls can occur. If the fall takes place in the bathroom, it will be detected by NWDs; conversely, if it occurs in another region of the house, the detection is made by WDs.

For the conduction of this study, we established the following goal for this research: *to assess the trade-off between different architectural arrangements of IoT-based FD systems using DEVS simulation models, WDs and NWDs*. We derived research questions (RQ) from this goal with the respective metrics and associated indicators, as presented next. The latter are pre-defined thresholds and an individual interpretation of a measure obtained through the metric.

RQ1: What is the accuracy delivered by each architectural arrangement?
Rationale: An IoT-based FD system can be designed using several types of sensing devices in the acquisition layer with different accuracy rates, as in Table II. By answering this question, we can assess the different global accuracy delivered by each one of the architectural arrangements under evaluation.
Metric: The metric used for answering this RQ is accuracy (**M1**), i.e., the percentage of falls accordingly detected by the architectural arrangement. As we used four datasets, M1 will be calculated for each dataset, and their average will be computed. Thus, we use the following equation for M1:

$$M1 = (WF + NWF)/DF \tag{1}$$

, where WF is the number of falls detected by the wearable device, NWF is the number of falls detected by the non-wearable device, and DF is the number of falls that the dataset contains. Indicators for accuracy are defined as follows: high if the total accuracy measured is greater than or equal to 90%; medium (from 70% to 90%); low (from 50% to 70%); and unreliable (less than 50%).

RQ2: What is the most cost-effective architectural arrangement?
Rationale: Given different older people's economic profiles interested in purchasing an FD system, they should be provided with several options with both different prices and detection ranges. This question guides researchers for offering, via simulation, objective measures to support designers and acquirers on the trade-off between cost and accuracy.
Metric: The metric developed to compare the architectural arrangements is **M2**: Cost (C) per Accuracy (M1), i.e., how expensive is to provide detection accuracy in each architectural arrangement. For M2, we use the following equation:

$$M2 = C/M1 \tag{2}$$

, where C is the total purchase cost of a given architectural arrangement, and M1 is the overall accuracy achieved by that same arrangement. Regarding cost per accuracy, we understand that we support a user of our model to trade-off about these measures, deciding the best architectural arrangement based on the available budget while preserving high accuracy. In this paper, we realize that a cost per accuracy greater than $5.00 can be considered expensive, while the other values can be regarded as cheap/valuable.
Experimental Frame. The tuple *{M1, M2}* then represents the output variables that compose the experimental frame under which the FD system was observed. We developed generators for injecting the data sets elaborated during data preparation, acceptors for checking on validity, and transducers for implementing the metrics.
Research instruments. The DEVS simulation specification formalism and MS4 Me IDE environment. MS4 Me supports the development of experimental frame components to implement modeling objectives [38], detailed herein in **Data Preparation**.
Computational model. The simulation model was automatically generated by MS4 Me IDE using the architectural design described in Section III-B.
Conditions of execution. The simulations ran on a laptop equipped with an Intel Core i7-5500U CPU @ 2.40GHz, with 8 GB of RAM Memory, HD of 500 GB, and running Ubuntu 18.04.2 LTS 64 bits.
Dataset selection criteria. The criteria for selecting datasets are as follows: the dataset content should contain both falls and activities of daily living (ADL); it should explicitly identify

the occurrence of falls, and the file format should be CSV (or easily convertible to CSV). Given these requirements, the datasets selected for this study were the Özdemir and Barshan dataset [39], the Gravity Project dataset [23], and the URFALL dataset [40].

Data preparation. We cleaned the datasets, so only the annotated column regarding the actions remained. The original Özdemir and Barshan[1] dataset is structured in several directories, each representing one actor and the actions he/she performed. We took each file representing one action and transferred it to an Excel spreadsheet. The resulting sheet contains approximately 16,000 different actions (states).

The URFALL dataset has 903 actions; however, many were repeated over time, i.e., several zeros represented a constant action. We gathered these repetitions in single steps, resulting in 90 different states, accurately as reported in the original study: 30 falls and 60 daily actions.

Finally, the Gravity Project dataset is specified in JSON. Then, we converted its content to CSV and the boolean values (true/false) into 0's and 1's. Initially, this dataset is partitioned into two pieces representing two different actors (GP1 and GP2): one performs 4,207 actions, whereas the other performs 2,754 actions. Hence, we consider four datasets in our experiments, not only three.

The four datasets date between 2014 and 2015 with real-world people; however, they include simulated falls since the subjects were actors. Hence, the purported falls do not represent actual falls (obtaining accurate fall data presents one of the most pressing challenges for research on FD systems). Also, those datasets are provided by wearable devices because of the lack of datasets sourced from non-wearable gadgets. In other words, our experiments used the four wearable-sourced datasets for both wearable and non-wearable devices.

B. Study Conduct

We performed thirty-six simulations, one per different combination of a dataset (OZBA, URFALL, GP1, and GP2) and an architectural arrangement (ARCH**1** to ARCH**9**).

We conducted the simulation for an entire day. Each simulation fed with OZBA took around 20 minutes, totaling 180 minutes. Due to the small size of the dataset, simulations with URFALL took approximately one minute each, spending nine minutes in total. Simulations with the GP1 and the GP2 datasets spent around 12 and 8 minutes each, respectively. Thus, the total time consumed with these datasets is 180 minutes. In brief, the simulation time was 369 minutes (or 6.15 hours).

For the delivery of data to the simulation, we applied the *Stimuli-SoS* approach [41]: we created artificial entities (the Human Body atomic model in Section IV-B) representing the surrounding environment to feed the simulation and run it.

C. Results

Results are discussed in a twofold manner: answering the analysis of the obtained data and subsequently answering the raised research questions.

Analyzing the Obtained Data. Table III presents the results from the simulation experiment executed with the OZBA dataset. The data represent the FD accuracy obtained for each architectural arrangement according to Equation 1, as well as the accuracy of each respective WD and NWD composing the arrangement.

TABLE III
THE OZBA DATASET SIMULATION RESULTS.

ID	Total accuracy	WD accuracy	NWD accuracy
ARCH**1**	**93.28%**	93.03%	95.91%
ARCH**2**	**92.89%**	92.60%	95.92%
ARCH**3**	**91.72%**	92.98%	79.82%
ARCH**4**	**95.43%**	95.20%	97.61%
ARCH**5**	**94.70%**	94.82%	93.58%
ARCH**6**	**93.78%**	95.33%	78.61%
ARCH**7**	**69.90%**	67.59%	93.27%
ARCH**8**	**71.37%**	69.23%	92.68%
ARCH**9**	**69.62%**	68.44%	81.48%

Observe that six architectural arrangements deliver a high accuracy, ranging from 91.72% (ARCH**3**) to 95.43% (ARCH**4**). It is noteworthy that the total accuracy is not the sum of the accuracy of the wearable and non-wearable devices because these are not in the same proportion, but instead as the proportion between the falls detected by both devices and the number of falls of a dataset (see Equation 1). For instance, the OZBA dataset includes 3,588 falls among all its actions. Thus, for ARCH**4**, the wearable device detected 3,097 falls, whereas the non-wearable device, 327 falls. In total, the architecture was capable of detecting 3,424 falls, which results in 95.43% of the total falls in the dataset.

Table IV presents the simulation results using URFALL. Again, ARCH**4** shows the highest accuracy, reaching a 100% accuracy. On the contrary, the ARCH**9** presents still the worst accuracy among the architectural alternatives analyzed.

TABLE IV
THE URFALL DATASET SIMULATION RESULTS.

ID	Total accuracy	WD accuracy	NWD accuracy
ARCH**1**	**90%**	88.46%	100%
ARCH**2**	**93.24%**	90%	0%
ARCH**3**	**90%**	92.31%	75%
ARCH**4**	**100%**	100%	100%
ARCH**5**	**96.67%**	96.55%	100%
ARCH**6**	**90%**	92.59%	66,67%
ARCH**7**	**70%**	60%	100%
ARCH**8**	**70%**	67.86%	100%
ARCH**9**	**53.33%**	51.72%	100%

Furthermore, we also observe a more significant difference in accuracy between the best and the worst arrangements as well as between the first-worst and second-worst ones if compared to the same analysis with the OZBA results. Using URFALL, there is a 46.67% difference between ARCH**4** and ARCH**9**, and 17% between ARCH**7** and ARCH**9**, respectively. That variation is of only 25.81% and 0.28%, respectively, using the OZBA dataset. We believe this discrepancy is because of the small size of the URFALL dataset, with only 30 falls represented.

[1]Throughout this paper, we refer to this dataset as OZBA.

By observing the accuracy of WDs and NWDs, we also find multiple values of 100%, e.g., ARCH**7** to ARCH**9**. These results are not a problem in the simulation, but also due to the size of the dataset. Therefore, we emphasize the relevance of choosing the dataset for in-silico experimentation with simulation to not impact on the results.

Now, consider the simulation results with the GP1 dataset, as shown in Table V. Likewise, the architectural arrangements with the best and worst accuracy values are ARCH**4** and ARCH**9**, respectively. In comparison with OZBA, there exists a small difference in accuracy between the best and the worst alternatives as well as between the first-worst and second-worst ones. Regarding the GP1 dataset, these values are 28.29% and 0.96%, respectively.

TABLE V
THE GP1 DATASET SIMULATION RESULTS.

ID	Total accuracy	WD accuracy	NWD accuracy
ARCH**1**	**93.20%**	92.74%	98.04%
ARCH**2**	**93.25%**	93.22%	93.57%
ARCH**3**	**91.84%**	93.01%	79.05%
ARCH**4**	**95.75%**	95.67%	96.62%
ARCH**5**	**95.52%**	95.35%	97.37%
ARCH**6**	**92.63%**	94.50%	74.23%
ARCH**7**	**68.42%**	65.43%	96.47%
ARCH**8**	**69.73%**	67.20%	96.69%
ARCH**9**	**67.46%**	66.50%	78.57%

Table VI shows the results of the simulation with the GP2 dataset. Besides ARCH**4**, the ARCH**5** alternative also reached the highest accuracy among all (i.e., 95.49%). On the other hand, ARCH**7** and ARCH**8** got accuracy values lower concerning the ARCH**9** arrangement, which presented the worst performance using the three previous datasets.

TABLE VI
THE GP2 DATASET SIMULATION RESULTS.

ID	Total accuracy	WD accuracy	NWD accuracy
ARCH**1**	**92.69%**	92.66%	92.91%
ARCH**2**	**92.75%**	92.64%	94.29%
ARCH**3**	**92.34%**	93.33%	80.70%
ARCH**4**	**95.49%**	95.40%	96.49%
ARCH**5**	**95.49%**	95.46%	95.74%
ARCH**6**	**92.69%**	94.02%	75.93%
ARCH**7**	**69.72%**	66.62%	98.59%
ARCH**8**	**69.72%**	67.47%	91.30%
ARCH**9**	**71.16%**	70.28%	80.65%

In comparison with OZBA, the difference in accuracy between the best and the worst alternatives are almost the same (i.e., 25.8%). Regarding the GP1 dataset, the variation of accuracy with the GP2 dataset is only 3.5% lower. Now, the difference between the first-worst and second-worst accuracy values using GP2 (ARCH**7** and ARCH**8**) reaches a higher but not so significant variation (i.e., 1.44%).

RQ1: What is the accuracy delivered by each architectural arrangement? Table VII presents the results from the simulation experiment. The data represent the FD accuracy for each different architectural arrangement and for each different dataset used. From all the accuracies obtained for each arrangement, we provided the average accuracy obtained from the simulation executions with each dataset used.

TABLE VII
AVERAGE ACCURACIES OF THE ARCHITECTURAL ARRANGEMENTS OF FD SYSTEMS.

ID	OZBA	URFall	GP1	GP2	Average accuracy
ARCH**1**	93.28%	90.00%	93.20%	92.69%	**92.29%**
ARCH**2**	92.89%	93.24%	93.25%	92.75%	**93.03%**
ARCH**3**	91.72%	90%	91.84%	92.34%	**91.48%**
ARCH**4**	95.43%	100%	95.75%	95.49%	**96.67%**
ARCH**5**	94.70%	96.67%	95.52%	95.49%	**95.60%**
ARCH**6**	93.78%	90%	92.63%	92.69%	**92.28%**
ARCH**7**	69.90%	70%	68.42%	69.72%	**69.51%**
ARCH**8**	71.37%	70%	69.73%	69.72%	**70.21%**
ARCH**9**	69.62%	53.33%	67.46%	71.16%	**65.39%**

From these data, we perceive the existence of two groups of architectural arrangements considering their average accuracies: ARCH**1** to ARCH**6** deliver higher values (91.48% to 96.67%), whereas ARCH**7** to ARCH**9** reach lower ones (65.39% to 70.21%). This is due to the difference in accuracy reported in the literature by the wearable and non-wearable devices in each architectural alternative.

On one hand, ARCH**4** and ARCH**5** reach a better performance because of the higher values of the accuracy of their wearable devices (95.29% and 92.7% in Table I, respectively). On the other hand, ARCH**7**, ARCH**8**, and ARCH**9** deploy the combination of a smartphone and a smartwatch as wearable equipment, which has the lowest accuracy among all devices (68% in Table I).

As 91% of elderly falls at home take place outside the bathroom [32], wearable devices in our simulation are more likely to be called upon to detect falls. This explains the fact that ARCH**4** reaches the best results using the four datasets, counting also on the highest accuracy of the non-wearable Doppler radar device (97% in Table I), yet in a smaller ratio.

RQ2: What is the most cost-effective architectural arrangement combining wearable and non-wearable devices? For answering this research question, we triangulated the accuracies obtained by answering **RQ1** with the purchase prices of each architectural arrangement (see Table II). We present the results in Table VIII.

TABLE VIII
COST PER ACCURACY OF THE ARCHITECTURAL ARRANGEMENTS FOR FD SYSTEMS.

ID	Average accuracy	Architectural cost	Cost per accuracy
ARCH**1**	92.29%	$5,528.99	**$59.91**
ARCH**2**	93.03%	$309.95	**$3.33**
ARCH**3**	91.48%	$148.94	**$1.63**
ARCH**4**	96.67%	$5,673.00	**$58.69**
ARCH**5**	95.60%	$453.96	**$4.75**
ARCH**6**	92.28%	$292.95	**$3.17**
ARCH**7**	69.51%	$5,733.00	**$82.48**
ARCH**8**	70.21%	$513.96	**$7.32**
ARCH**9**	65.39%	$352.95	**$5.40**

By analyzing the cost per accuracy, we offer to an FD system designer and clients the possibility to decide on an arrangement to be acquired based on their requirements, needs, and financial conditions. We observe the following findings:

- the most expensive architecture (ARCH**7**) delivers a low accuracy, but it uses WDs that many people already have, such as smart watches and smartphones;
- although the Doppler radar kit is an expensive device, it is broadly accepted [42], since it does not hamper the user privacy (as other non-wearable devices do, e.g., cameras);
- ARCH**4** delivers the greatest accuracy and is one of the most expensive options (Doppler radar and smartphone);
- in contrast, ARCH**3**, which delivers roughly 5% less of accuracy in comparison with ARCH**4**, still presents a high performance for FD (91.48%), and has an acquisition price 36 times cheaper than ARCH**4**.
- Analyzing the architectures that have the lowest costs (ARCH**2**-**3**, ARCH**5**-**6**, and ARCH8-9), we can observe that ARCH**5** offers the greatest precision.

Therefore, ARCH**3** is the cheapest architectural arrangement. However, with only 3 times its value, ARCH**5** has 4% more accuracy, making it the most cost-effective architectural arrangement combining WD and NWD for FD purposes.

VI. Discussion

FD systems are critical because they are directly related to saving people's lives. They must be tested and demonstrate excellent results in terms of reliability before being deployed. However, real-world testing can be cumbersome because it usually requires a lot of resources and time. Besides, as fall simulations are necessary, another difficulty is in recruiting real users of these systems to test them in a safe manner [1].

In this paper, we seek to encourage the use of *in silico* simulations to test FD systems. This type of simulation has been successfully used in several disciplines, such as Economy, Biology, Social Sciences, and also in Software Engineering, to support the visualization of the dynamic structure and systems behaviors [19], [43]. Simulations can anticipate, at design time, failures, and behaviors that could potentially occur at run time and is a fundamental tool to understand physical phenomena in the field of Engineering Sciences. It is even more critical in the case of Healthcare Engineering as it can help in reducing the number and duration of the necessary field experiments; these are costly and difficult to conduct [10].

We also highlight the importance of the datasets for the study we conducted. We adopted four datasets with different dimensions, varying from 90 to 16,000 actions (ADLs and falls). We showed that the use of small datasets can deliver results that are not fully trustworthy since they can reach a false 100% accuracy, as delivered by arrangement ARCH**4** using *URFALL*. ARCH**9** also presents an uncommon result because its accuracy is much lower than the others. Therefore, we perceive the influence of the datasets on the experiment and recommend the use of a diverse group of distinct datasets for obtaining more reliable results.

Our simulation development differs from others also in its use of the SES to support composability. The composability feature of SES results in significant reduction in time to develop models for new objectives [15], [44]. Since we could reuse the wearable and non-wearable models for specifying the nine different variants of the architecture, we perceived an important time reduction in our project due to the adoption of SES structure and relying on MS4Me infrastructure. This can represent an important reduction in total development time when compared to other simulation formalisms or tools.

A. Limitations and Threats to Validity

In this section, we discuss the main limitations and threats that should be taken into account when analyzing the results and findings obtained herein.

1) Limitations of the Study: As with any research endeavor, our study has limitations to be considered.

- **External factors:** We do not address aspects that interfere with conventional's or depth camera's accuracies, such as lighting and occlusions by objects. Another factor not considered is people's privacy due to cameras in the bathroom. This explains why Doppler radar has been given special attention in recent literature.
- **Users adherence:** We assume that elderly people wear the devices even when they are outside the bathroom. We do not deal with a lower detection rate due to a non-adherence of the user, i.e., in case a fall happens outside the bathroom, and the person is not wearing the device.
- **Error-free sensing layer:** Should a fall be detected, we consider that all fall-related raw data is genuine and not subject to errors.
- **Reasoning algorithms:** We are not concerned with reasoning algorithms for FD interpretation [45]. As the four datasets already annotate the instant of a fall, we consider the accuracy of sensing devices in the acquisition layer, as reported in the literature.
- **Latency:** For system's efficacy, the set of techniques for FD, mostly in the acquisition, reasoning, and dissemination layers, may well be delay-sensitive. Moreover, datasets generally do not provide such information.

2) Threats to Validity: Threats to validity can be of four types [46]: conclusion, internal, construction, and external. As threats to **conclusion validity**, we cite those related to the establishment of a statistical relation between the individual arrangements and the total arrangement. We mitigate that threat with mathematical functions used to associate the individual accuracies with the global accuracies. Hence, the procedure is auditable and repeatable.

For **internal validity**, we mention simulation correctness and datasets selection. The correctness of simulation is a constant risk, i.e., how precise is the representation and how correct is its implementation. To relieve this threat, we continuously checked whether the (i) number of inputs was proportional to the number of outputs, and (ii) the accuracy delivered by each architectural arrangement was close to the average of the accuracies delivered by the devices involved in each arrangement. We also submitted the simulation code to a healthcare system expert so that he could evaluate how precise is the representation in regards to the intended system. He agreed that the simulation accurately represents FD systems. Moreover, for diversifying the samples and obtaining strength of evidence, we used four datasets obtained from different providers with different dimensions, ranging from 90 to 16,382 actions. This prevents an inherent bias due to the use of only one or two specific (and similar) datasets, which potentially would deliver similar results. Results were significantly different in each architectural arrangement for the different datasets.

About **construction validity**, we cite threats related to the FD precision in each device represented and the extent of the FD system represented. For mitigating possible risks associated with the former, we adopted accuracies reported in studies of the specialized literature. On the latter, we focus only on the IoT acquisition layer of the FD system.

As the modeling, reasoning, and dissemination layers are not represented, our conclusions can not be generalized to these layers, which leads to the threat to **external validity**. Our findings are only related to the acquisition layer and consider the specific architectural arrangements simulated. However, our approach has the potential to be generalized as a technique to predict other different architectural arrangements as well as supporting a trade-off between accuracy and purchase prices.

VII. Conclusion

Simulation models have been applied in the literature to specific domains to predict real systems' properties and assess specific attributes to support decisions about systems architectures [16], [20]. Similarly, our study adopts simulation models to evaluate IoT architectures concerned with FD as a simulation application scenario.

Our work mainly contributes with a DEVS-encoded simulation model for FD systems supporting, at design time, the trade-off between cost and accuracy of multiple architectural arrangements based on a diverse configuration of sensing mechanisms in the acquisition layer (i.e., accelerometers, depth cameras, and radar). We conducted a robust six-hour simulation study with thirty-six different configurations, involving nine architectural arrangements and four datasets widely accepted in FD systems literature. Combining those efforts with analyzing the study's threats to validity reinforces the strength of evidence of the achieved results.

Although our work is focused on design time, we envision the possibility of connecting already deployed systems to simulation models to raise what is currently known as digital twins. Future work can investigate the adoption of DEVS models in digital twins infrastructure so that we can integrate detection of sudden changes in acceleration and maybe infer, by simulation, that a fall is about to happen. Further procedures can then be implemented, such as preventive alerts or robotic apparatus to avoid imminent falls.

Acknowledgment

The authors thank MS4 Systems Inc. (RTSync) for partially funding the conduction and publication of this work besides making available a free academic license of the MS4 Me modeling and simulation integrated development environment.

References

[1] S. Chaudhuri, H. Thompson, and G. Demiris, "Fall detection devices and their use with older adults: a systematic review," *Journal of Geriatric Physical Therapy*, vol. 37, no. 4, pp. 178–196, 2014.

[2] E. Casilari, J.-A. Santoyo-Ramón, and J.-M. Cano-García, "Analysis of public datasets for wearable fall detection systems," *Sensors*, vol. 17, no. 7, pp. 1–28, 2017.

[3] M. Mercuri, P. J. Soh, G. Pandey, P. Karsmakers, G. A. E. Vandenbosch, P. Leroux, and D. Schreurs, "Analysis of an indoor biomedical radar-based system for health monitoring," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 5, pp. 2061–2068, 2013.

[4] Y. D. Zhang and H. Lan, "Bathroom fall detection based on ultrasonic doppler effect," *Journal of Shanghai Normal University (Nature Sciences)*, vol. 47, no. 2, pp. 225–229, 2018.

[5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[6] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, and V. H. C. de Albuquerque, "A reference model for internet of things middleware," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 871–883, 2018.

[7] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, and S. Zhang, "Recommendations on the internet of things: Requirements, challenges, and directions," *IEEE Internet Computing*, vol. 23, no. 3, pp. 46–54, 2019.

[8] P. Ntanasis, E. Pippa, A. Øzdemir, B. Barshan, and V. Megalooikonomou, "Investigation of sensor placement for accurate fall detection," in *Proceedings of the 6th International Conference on Wireless Mobile Communication and Healthcare*, pp. 225–232, 2017.

[9] B. B. N. de França and G. H. Travassos, "Simulation based studies in software engineering: A matter of validity," *CLEI Electron. J.*, vol. 18, no. 1, 2015.

[10] N. Noury and T. Hadidi, "Computer simulation of the activity of the elderly person living independently in a health smart home," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 3, pp. 1216 – 1228, 2012.

[11] L. Gutiérrez-Madroñal, L. L. Blunda, M. Wagner, and I. Medina-Bulo, "Test event generation for a fall-detection iot system," *IEEE Internet of Things Journal*, vol. 6, pp. 6642–6651, 2019.

[12] G. Mastorakis, T. J. Ellis, and D. Makris, "Fall detection without people: A simulation approach tackling video data scarcity," *Expert Syst. Appl.*, vol. 112, pp. 125–137, 2018.

[13] A. Makhlouf, I. Boudouane, N. Saadia, and A. Ramdane-Cherif, "Ambient assistance service for fall and heart problem detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, p. 1527–1546, 2018.

[14] J. T. Bruno and R. F. Bulcão-Neto, "Simulação de arquiteturas de sistemas de monitoramento de quedas de pacientes domiciliares," in *Proceedings of the Workshop on Modelling and Simulation of Software-Intensive Systems, MSSiS 2019, Brazil*, pp. 420–433, SBC, 2019.

[15] B. P. Zeigler, H. Praehofer, and T. Kim, *Theory of Modeling and Simulation*. Cambridge, Massachusetts: Academic Press, 2nd ed., 2000.

[16] R. Gomes Alves, R. Filev Maia, and F. Lima, "Discrete-event simulation of an irrigation system using internet of things," *IEEE Latin America Transactions*, vol. 20, no. 6, p. 941–947, 2022.

[17] E. Cippitelli, F. Fioranelli, E. Gambi, and S. Spinsante, "Radar and rgb-depth sensors for fall detection: A review," *IEEE Sensors Journal*, vol. 17, no. 12, pp. 3585–3604, 2017.

[18] M. E. Karar, H. I. Shehata, and O. Reyad, "A survey of iot-based fall detection for aiding elderly care: Sensors, methods, challenges and future trends," *Applied Sciences*, vol. 12, no. 7, 2022.

[19] B. B. N. d. França and G. H. Travassos, "Are We Prepared for Simulation Based Studies in Software Engineering Yet?," *CLEI Electronic Journal*, vol. 16, pp. 9 – 9, 2013.

[20] M. J. Blas and S. Gonnet, "Using model-to-model transformations for web software architecture simulation," *IEEE Latin America Transactions*, vol. 20, no. 4, p. 545–552, 2021.

[21] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive Mob. Comput.*, vol. 6, no. 2, pp. 161–180, 2010.

[22] E. Casilari and M. A. Oviedo-Jiménez, "Automatic fall detection system based on the combined use of a smartphone and a smartwatch," *PloS one*, vol. 10, no. 11, p. e0140929, 2015.

[23] T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lønes, J. N. Slettevold, and S. M. Weggersen, "A combined smartphone and smartwatch fall detection system," in *2015 IEEE International Conference on Computer and Information Technology; Ubiq-*

uitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1443–1448, 2015.

[24] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. Ngu, and C. C. Rivera, "Smartfall: A smartwatch-based fall detection system using deep learning," *Sensors*, vol. 18, no. 10, p. 3363, 2018.

[25] A. Ngu, Y. Wu, H. Zare, A. Polican, B. Yarbrough, and L. Yao, "Fall detection using smartwatch sensor data with accessor architecture," in *International Conference on Smart Health*, pp. 81–93, Springer, 2017.

[26] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, 2019.

[27] A. Naeem and S. Khan, "Fall detection using accelerometer calibration," in *2019 15th International Conference on Emerging Technologies*, pp. 1–6, 2019.

[28] M. Skubic, B. H. Harris, E. Stone, K. Ho, B.-Y. Su, and M. Rantz, "Testing non-wearable fall detection methods in the homes of older adults," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 557–560, 2016.

[29] L. Anishchenko, A. Zhuravlev, and M. Chizh, "Fall detection using multiple bioradars and convolutional neural networks," *Sensors*, vol. 19, no. 24, p. 5569, 2019.

[30] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, "Depth-based human fall detection via shape features and improved extreme learning machine," *IEEE journal of biomedical and health informatics*, vol. 18, no. 6, pp. 1915–1922, 2014.

[31] B. Erol, M. Francisco, A. Ravisankar, and M. Amin, "Realization of radar-based fall detection using spectrograms," in *Compressive Sensing VII: From Diverse Modalities to Big Data Analytics*, vol. 10658, p. 106580B, International Society for Optics and Photonics, 2018.

[32] D. Wild, U. S. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?," *Br Med J (Clin Res Ed)*, vol. 282, no. 6260, p. 266–268, 1981.

[33] C. Medrano, R. Igual, I. Plaza, and M. Castro, "Detecting falls as novelties in acceleration patterns acquired with smartphones," *PloS one*, vol. 9, no. 4, p. e94811, 2014.

[34] B. P. Zeigler, C. Seo, and D. Kim, "System entity structures for suites of simulation models," *International Journal of Modeling, Simulation and Scientific Computing*, vol. 4, no. 03, pp. 13400061–134000611, 2013.

[35] B. P. Zeigler and H. S. Sarjoughian, *Guide to modeling and simulation of systems of systems*. Springer, 2017. 2nd Edition.

[36] A. in https://ww2.inf.ufg.br/~insight/projects/falldetection/.

[37] B. B. N. de França and G. H. Travassos, "Experimentation with dynamic simulation models in software engineering: planning and reporting guidelines," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1302–1345, 2016.

[38] B. P. Zeigler, A. Muzy, and E. Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic Press Elsevier, 2018.

[39] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, 2014.

[40] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer methods and programs in biomedicine*, vol. 117, no. 3, pp. 489–501, 2014.

[41] V. V. Graciano Neto, C. E. B. Paes, L. Garcés, M. Guessi, W. Manzano, F. Oquendo, and E. Y. Nakagawa, "Stimuli-sos: a model-based approach to derive stimuli generators for simulations of systems-of-systems software architectures," *Journal of the Brazilian Computer Society*, vol. 23, no. 1, pp. 1–22, 2017.

[42] B. Y. Su, K. Ho, M. J. Rantz, and M. Skubic, "Doppler radar fall activity detection using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 3, pp. 865–875, 2014.

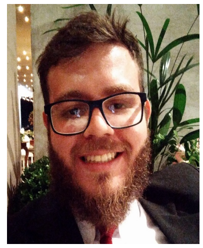
[43] W. Manzano, V. V. Graciano Neto, and E. Y. Nakagawa, "Dynamic-sos: An approach for the simulation of systems-of-systems dynamic architectures," *The Computer Journal*, 2019.

[44] B. Zeigler, N. Keller, D. Kim, and C. Anderson, "Supporting the reuse of algorithmic simulation models," in *Proceedings of the 2020 Summer Simulation Conference*, (Virtual Conference), pp. 1–12, SCS, 2020.

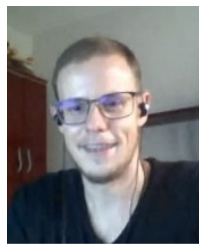
[45] A. Ramachandran and A. Karuppiah, "A Survey on Recent Advances in Wearable Fall Detection Systems," *BioMed Research International*, vol. 2020, p. 2167160, 2020.

[46] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

Renato F. Bulcão-Neto received his Ph.D. degree in Computer Science and Computational Mathematics from the University of São Paulo, São Carlos, Brazil (2006). He has held a Tenured-Position as a Professor at the Federal University of Goiás (UFG), Brazil, since 2010. His research interests include software requirements engineering, the Internet of Things, and health informatics. Contact him at rbulcao@ufg.br.



Paulo Gabriel Teixeira received his MSC. degree in Computer Science from the Federal University of Goiás (UFG), Brazil (2021). His research interests are software architecture, systems simulation, and systems-of-systems. Currently, he works at SiDi, Campinas, as a software developer. Contact him at paulogabriel.teixeira@gmail.com.



Bruno G. Araujo Lebtag received his MSC. degree in Computer Science from the Federal University of Goiás (UFG), Brazil (2021). His research interests are software architecture, systems simulation, and systems-of-systems. Contact him at brlebtag@gmail.com.

Valdemar V. Graciano Neto received his Ph.D. double-degree in Computer Science and Computational Mathematics from the University of São Paulo, São Carlos, Brazil, and Informatics from IRISA Labs at the University of South Brittany, Vannes, France (2018). He has held a Tenured-Position as a Professor at the Federal University of Goiás (UFG), Brazil, since 2014. His research interests include software architecture, modeling and simulation, and systems-of-systems. Contact him at valdemarneto@ufg.br.

Alessandra A. Macedo is an Associate Professor for the Department of Computing and Mathematics, FFCLRP, University of São Paulo (USP), Ribeirão Preto, SP, Brazil, since 2004. Current research interests include health informatics, computer-aided clinical decision support systems, information retrieval, software engineering, information extraction, and knowledge representation. Contact her at ale.alaniz@usp.br.

Bernard P. Zeigler is Chief Scientist for RT-Sync Corp and Professor Emeritus of Electrical and Computer Engineering at the University of Arizona, Tucson. Dr. Zeigler's Ph.D. is in Computer/Communication Science from University of Michigan, Ann Arbor, MI (1968). His research interests include methodology of modeling and simulation, modeling and simulation of healthcare systems, model-based system engineering, and software/hardware support for M&S Development. He has published numerous books and research papers on the DEVS formalism. Zeigler is recognized as a Computer Simulation Pioneer and internationally known for his 1976 foundational text Theory of Modeling and Simulation, revised for a third edition (Academic Press, 2018). Contact him at zeigler@rtsync.com.