ADVANCED MIGRAINE PREDICTION SIMULATION SYSTEM

Josué Pagán

Department of Computer Architecture and Automation
Complutense University of Madrid
Computer Science Faculty, Spain
CCS-Center for Computational Simulation
Madrid, Spain
jpagan@ucm.es

José M. Moya

LSI - Electronic Engineering Department
Technical University of Madrid, Spain
CCS-Center for Computational Simulation
Madrid, Spain
josem@die.upm.es

José L. Risco-Martín

Dept. of Computer Architecture and Automation
Complutense University of Madrid
Computer Science Faculty, Spain
jlrisco@ucm.es

José L. Ayala

Dept. of Computer Architecture and Automation
Complutense University of Madrid
Computer Science Faculty, Spain
jayala@ucm.es

ABSTRACT

In the Internet of Things (IoT) era, there is growing interest in wireless monitoring sensors for detection, classification and prediction of health symptoms. The prediction of symptoms in chronic diseases such as migraines brings new hope to improve patients' lives. The prediction of a migraine symptomatic event through monitoring hemodynamic variables has been previously demonstrated in our earlier work. In this paper, a simulation-based approach for a real-time migraine prediction system is described. The simulation has been implemented using the specifications of the formal description language Discrete EVent Systems (DEVS). The simulation system is a proof of concept that is ready for testing in a real-world ambulatory monitoring environment. The results obtained encourage developing a hardware/software (HW/SW) co-simulation system that incorporates Hardware-in-the-Loop (HIL) components as prior step to the expensive and slow hardware implementation of a complete migraine prediction device. When such a system is used in a real-time setting, it can simulate failures in sensors and trigger alarms for active patient response.

Keywords: migraine prediction, failure detector, robust system.

1 INTRODUCTION

Recent proliferation of wireless monitoring devices has brought big opportunities to the industry of healthcare and personal well-being. This has become a major concern in the paradigm of proactive personal eHealth (Zheng et al. 2014) in the Internet of Things (IoT) era. Predictive models in the eHealth scenario using wearable monitoring devices have increased rapidly—mostly oriented to activity recognition (Barshan 2014). Unfortunately, event detection in neurological diseases and diagnosis are still in their early stages of research and with limited commercial examples.

Development of devices for diagnosis and detection is a time consuming process. Simulation experiments may help speed up the engineering process, especially in the initial phase of exploration. This is the main goal of the paper: to specify an advanced simulation framework that helps to validate the behavior of a migraine attack predictive system.

The migraine is a neurological disease that causes strong headaches. It is considered one of the most disabling neurological diseases and affects around 10% of population worldwide (Lipton and Scher 2001) and 15% in Europe (Stovner and Andree 2010). The migraine causes fatigue, anxiety or cardiovascular problems. The migraine worsens the patients' life and their performance at work or school, which lead to high cost for private and national health systems. On a long-term basis, it is a social problem. Estimated direct and indirect costs reach €1,222 per patient per year in Europe (Linde et al. 2012). A migraine prediction system with 76% accuracy and massively deployed only in 2% of European migraine sufferers shows a potential savings of € 1272 million due to the benefits of the migraine prediction (Pagán et al. 2017).

A cascade of neurological processes precede a migraine followed by the pain for the next few hours or days. Some migraine sufferers experience symptoms that may occur from three days to hours before the pain starts (Giffin et al. 2003). These symptoms are called premonitory or prodromic symptoms and they are subjective and unspecific: nausea, yawns, tearing, *etc.* Some patiens also suffer from auras. Auras are objective and specific disturbances such as losing vision that occurs commonly within 30 minutes before the onset of pain.

Pharmacokinetics defines the mechanisms of absorption and distribution of substances in an organism. Because of the pharmacokinetics of current drugs for treatment of migraine in the accute phase, prodromic symptoms and auras—some times—are not helpful to stop the pain—as it is difficult to estimate the onset of pain. Most migraine sufferers wait for the interval between period pain episodes to take the specific medication. The delayed intake reduces the effectiveness of the treatment. Thus, prediction of the onset of a migraine attack will help the patient to stop the pain.

The Autonomous Nervous System (ANS) regulates body conditions through blood circulation (blood flow) at adequate rate. This lead to changes in the hemodynamic variables. When a migraine occurs, changes appear in these variables. Prediction modeling of migraines has been demonstrated by the authors in previous works (Pagán et al. 2015, Pagán et al. 2016). In these works the predictive modeling of the migraine symptoms was shown feasible through the analysis of the changes in four hemodynamic variables controlled by the ANS: skin temperature (TEMP), electrodermal activity (EDA), oxygen saturation (SpO2) and heart rate (HR). These two works addressed the problem through the use of classic modeling methods like state-space systems in (Pagán et al. 2015) and heuristics like Grammatical Evolutionary (GE) algorithms in (Pagán et al. 2016).

Once the offline (e.g. in virtual, stand-alone mode) predictive modeling has been demonstrated, the next step is to test it in real-time. The first step, prior to the expensive and slow hardware implementation of a complete prediction and monitoring device, is the simulation of our migraine prediction system. The simulation system, in order to be useful, must be able to raise an alarm and alert patients. The advanced simulator presented in this work simulates a robust system against sensor failures that performs error signal detection and signal recovery. In case that the sensors are not available, it executes a hierarchical methodology of predictive models selection if signal recovery is not possible. A co-designed monitoring device has been developed in collaboration with the company M2C[1] but this device does not raise alarms or performs predictions. Before an actual device is implemented in hardware, a hardware/software (HW/SW) co-simulation that includes hardware-in-the-loop (HIL) will be used. This will ensure that the system works in presence of actual hardware sensor failures and physical actuators, and triggers alarms

[1] http://www.m2csolutions.com/

accurately, as predicted by the simulation system. The specification of such a HW/SW co-simulation system is specified using the Discrete Event Systems (DEVS) formalism (Zeigler et al. 2000) that specifies unambiguous structure and behavior of any hybrid complex system. This is an incremental design with easy component substitution and rapid HW/SW swapping mechanism as previously shown through a DEVS-based transparent HW/SW modeling and simulation framework in (Risco-Martín et al. 2016). In this paper we implement a DEVS-based model that will be the basis for the aforementioned HIL system. With this, we will be able to start clinical experiments to inform patients when to take medications in advance followed by a study of the benefits of prediction in terms of complete or partial pain relief.

The paper is organized as follows: specifications on DEVS formalism and state of the art in simulation is described in Section 2. The developed advanced simulation framework developed is shown in Section 3; where simulator and its parameters are shown. Finally, Section 4 shows the evaluation of the simulator and its validation for future analyses. Finally, the paper is concluded in Section 5.

2 BACKGROUND

What we pursue in this paper is a simulation of a Cyber-Physical System (CPS) to raise alarms for predictive modeling of symptomatic crises in chronic diseases, specifically, the migraine. To the best of our knowledge, this study is the first attempt to simulate a real device for the prediction of symptomatic crises.

Concerning simulation frameworks for simulations of CPSs we can find Ptolemy II (Buck et al. 1994), a discrete-event modeling environment focused on application to cyber-physical and embedded systems; or Simulink from MATLAB [2], that is more oriented towards engineers and has hard semantics. Barhak *et al.* (2010) present a software tool for chronic diseases. Despite of the software is presented as a tool for many different chronic diseases, models must be defined as states and transition probabilities of Markov transition models—which is a hard constraint. Another example is Archimedes (Eddy and Schlessinger 2003), a commercial simulator for diabetes using an object-oriented approach. Both of them implement a Graphical User Interface (GUI) and are distributed as open source.

DEVS modeling and simulation formalism allows to simulate models under a unified modeling and simulation theory in real time, soft-real time and virtual time, which makes DEVS a good choice for a formal validation before the real implementation of a physical device. The absence of a formal backup like DEVS in all of the three simulators aforementioned can be used to perform this study.

We decided to use xDEVS—published as Open Source under General Public License (GPL) in[3] —because of the nature of the different predictive models we can use (based on GE or state-space algorithms). Furthermore, its semantics, performance, and its implementation into a hardware device is very straight forward, due to xDEVS is coded in JAVA. Despite this simulation environment does not implement a GUI yet, we are working on the implementation of a Unified Modeling Language (UML) executable interface as described in (Risco-Martín et al. 2009).

DEVS is a modular and hierarchical modeling formalism, with all of the advantages and uses of simulation systems, such as: completeness, verifiability, extensibility, and maintainability and allows execution of Monte Carlo simulations, parallel simulation using threads or distributed using webs (Wainer et al. 2008), as an example. In this paper we use the xDEVS open source JAVA library with the aim of making a future implementation on a hardware device. DEVS is a general formalism for discrete event system modeling based on a mathematical Set Theory (Zeigler et al. 2000). Over the last four decades it has been used to implement a formally described system using an existing software/hardware library in multiple languages (e.g. Lisp, Scheme, C++, JAVA, Python, *etc.*). There are two types of models in DEVS: atomic and coupled.

[2]MATLAB 2015. version 8.5.0.197613 (R2015a). Natick, Massachusetts, The MathWorks Inc.

[3]xDEVS 2016: JAVA library for DEVS. https://github.com/jlrisco/xdevs

An atomic model is irreducible and it specifies the behavior for any modeled entity: processes an input event based on its state and condition, and generates an output event and changes its state. DEVS formally represents an atomic model by three sets: input (X), output (Y) and state (S), and five functions: time advance (ta), external transition (δ_{ext}), internal transition (δ_{int}), confluent (δ_{con}) and output (λ). Formally, it is expressed as follows:

$$A = \langle I, O, X, S, Y, \lambda, \delta_{\text{int}}, \delta_{\text{ext}}, \delta_{\text{con}}, ta \rangle \tag{1}$$

where:

- I is the set of input ports.
- O is the set of output ports.
- X is the set of inputs described in terms of pairs port-value: $\{p, v\}$.
- S is the state space. It includes the current state of the atomic model and also two special parameters called σ and *phase*. σ is the time until the next event generation, and the *phase* is a description of the current state (usually in natural language).
- Y is the set of outputs, also described in terms of pairs port-value: $\{p, v\}$.
- $\lambda : S \rightarrow Y$ is the output function. When the time elapsed since the last output function is equal to σ, then λ is automatically executed.
- $\delta_{int} : S \rightarrow S$ is the internal transition function. It is used to change the state S, *phase* and σ, and it is executed right after the output function (λ).
- $\delta_{ext} : Q \cdot X^b \rightarrow S$ is the external transition function. It is automatically executed when an external event arrives to one of the input ports, changing the current state if needed.
  - $Q = (s, e), s \in S, 0 \leq e \leq ta(s)$ is the total state set, where e is the time elapsed since the last transition.
  - X^b is a bag of elements of X.
- $\delta_{con} : Q \cdot X^b \rightarrow S$ is the confluent function, subject to $\delta_{con}(s, \emptyset) = \delta_{int}(s)$. This transition is selected if δ_{ext} and δ_{int} must be executed at the same instant.
- $ta(s) : S \rightarrow \Re_0^+ \cup \infty$ is the time advance function.

A coupled model aggregates and interconnects two or more atomic or coupled models. And it is formally described as:

$$M = \langle I, O, X, Y, C_i, EIC, EOC, IC \rangle \tag{2}$$

where:

- I, O are the set of external (not coupled) input and output ports.
- X is the set of external input events.
- Y is the set of output events.
- C_i is a set of DEVS component models (atomic or coupled). Note that C_i makes this definition recursive.
- EIC is the external input coupling relation.
- EOC is the external output coupling relation.
- IC is the internal coupling relation.

Due to the definition in Eq. 2, a coupled model can itself be a part of a component in a larger coupled model system giving rise to a hierarchical DEVS model construction.

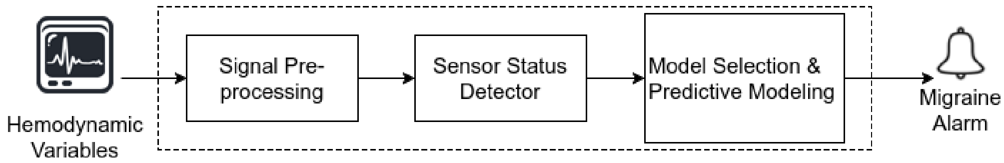


Figure 1: Conceptual system diagram. The dotted block represents the system.

The earlier developed GE models in (Pagán et al. 2016) was extended to validate the proof of concept of migraine prediction modeling using the DEVS-based approach in our other work (Pagán et al. 2016). A top-down view of the simulator is drawn.

3 ADVANCED MIGRAINE PREDICTION SYSTEM

In this section, a top-down view of the migraine prediction system is described.

3.1 Conceptual model

In the basic version (Pagán et al. 2016), we demonstrated that GE algorithms can be used to predict a migraine using simulation by raising an alarm. The mechanism was based on a simple threshold monitoring agent. In this work, we used state-space algorithms using the formal specifications as required in a DEVS-based simulator. This brings formal rigor to the modeling effort.

In the current version of our migraine prediction simulator which is based on our earlier developed simulator (Pagán et al. 2015), we aim to demonstrate a robust methodology against sensor failures. We showed that if the monitoring and prediction systems detects anomalies in sensors, a Sensor Dependant Model Selection System ($SDMS^2$) can choose an appropriate set of models that avoids the use of a damaged sensor and maintain an accuracy level for a given prediction horizon. Prior to changing the set of models, the system computes statistical averages to estimate lost sensor's values. If the failure in sensor exceeds a pre-defined wall clock interval, the $SDMS^2$ will choose a different set of models. Figure 1 represents a conceptual diagram of the current approach. *Hemodynamic variables* and *Migraine alarm* represent sensors and a reactive device respectively, and can be easily replaced for HIL implementation. The *Sensor Status Detector* monitors the sensors operations and makes decisions if operations are below a specified threshold. *Model Selection* selects the set of operational sensors that continue to meet the operational requirements.

Figure 2 represents the top view of the advanced migraine predictor simulator system. Most blocks are described in detail in our previous works (Pagán et al. 2015) and (Pagán et al. 2016). A detailed explanation is not necessary for a comprehensive understanding of our current research, but main specifications are shown in sections ahead. Shadowed boxes in Figure 2 represent coupled models, and there are seven of them divided into five types. Atomic models are represented with uncoloured boxes, and there are twenty-five top-level atomic models divided into seven different types. Models surrounded by dotted lines are not part of the migraine prediction system, but they are required in the simulation framework. These will be removed in a real implementation of the system.

3.2 DEVS formalization of the conceptual model

We now describe the detailed migraine prediction system using the DEVS formalism.

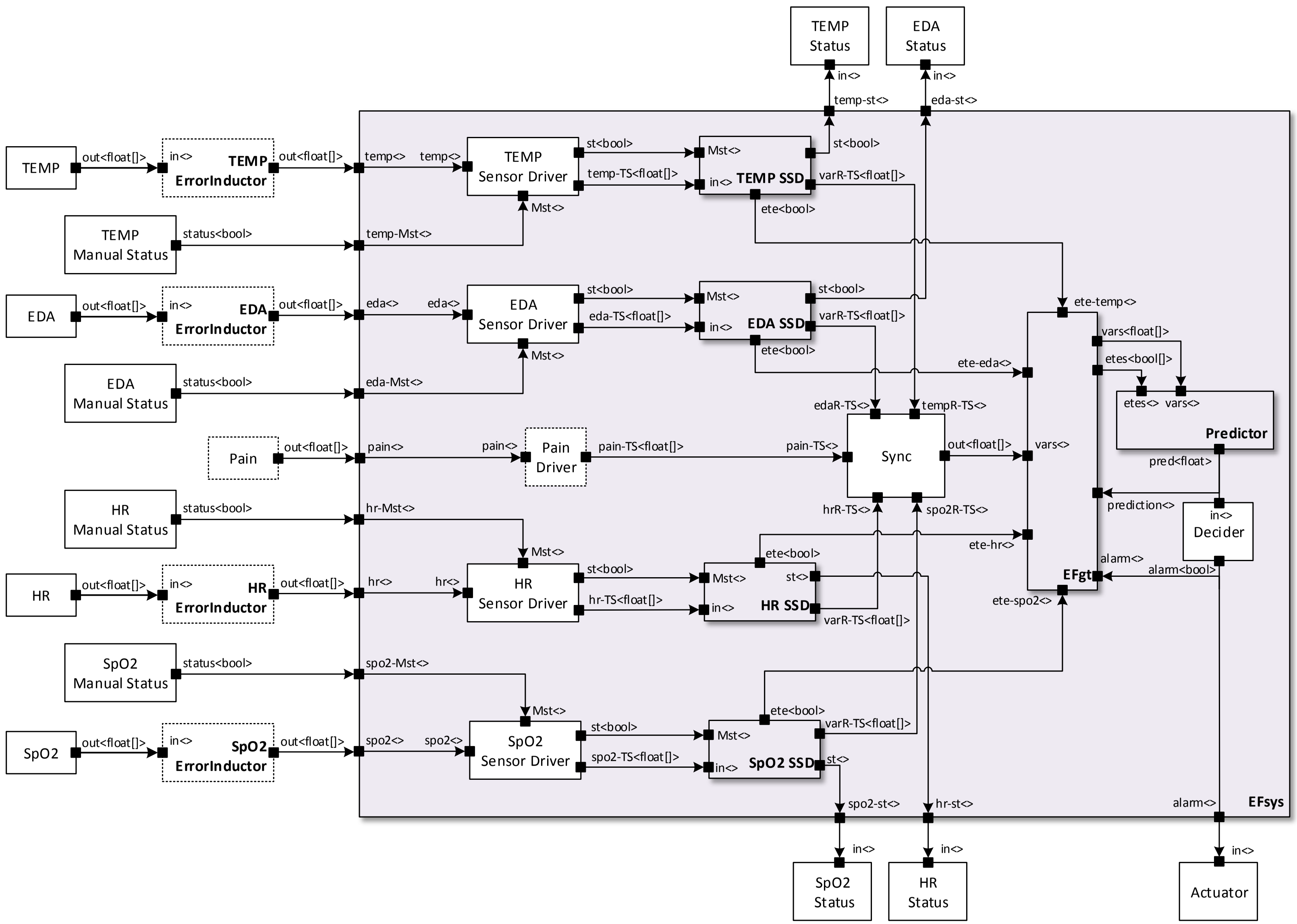


Figure 2: The `RootCoupled` model is the simulator framework containing all the coupled and atomic models in the system.

3.2.1 Coupled models

The five different types of coupled models in the system are the `EFsys`, the `EFgt`, the `SSD`, the `Predictor` and the `Graphs`. As the `RootCoupled` is the actual system, it is the entire Figure 2 itself. The coupled model `RootCoupled` is the simulator frame and interconnects the models that simulate the hardware modules, with the prediction system model in `EFsys`.

The `EFsys` (in Figure 2) hosts the remaining coupled models detailed in Figures 3a through 3c. The `EFsys` contains the intelligence of the system, which cannot be replaced by hardware. It also performs the processing from data of the single-output atomic models `TEMP`, `EDA`, `HR` and `SpO2` after data pass through the error sources `ErrorInductor`. The system also has as inputs other single-output atomic models named as `Manual Status` for each sensor, and gives an output to the single-input atomic models `Actuator`, and `Status` for each sensor as well.

The `EFgt` model (Figure 3a) contains two atomic models to control the data flow through model `G`, and to show simulation statistics through model `T`. If the simulator runs in simulated time, `T` activates the *stop* signal to finish the simulation after the simulation's observation time has elapsed.

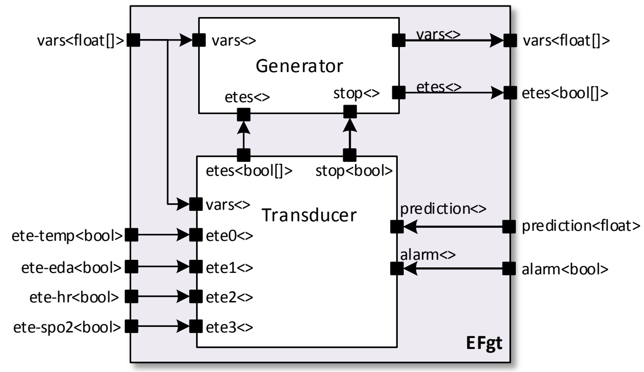
Our simulator also has a coupled model (`Graphs`) that does not appear in Figure 2 for the sake of simplicity. `Graphs` has been included to improve the user experience. This model is only suitable for software simulation and plots the input data, the migraine predictions, the status of sensors, the manual resets of sensors and, the alarm event if it occurs.

`SSD` is the acronym of Sensor Status Detector (see Figure 3b). This coupled model is able to detect three types of abnormal behaviors in signals: noisy signal, disconnection of a sensor (fall) and saturation. When an anomaly is detected, the `Anomaly Detector` raises an alarm signal `detect<bool>`. This alarm will indicate that the sensor has a problem and the `GPML` atomic model gets activated to recover the signal based on recent buffered data. The `GPML` model performs a Gaussian Process Machine Learning (Rasmussen and Williams 2005) as explained in (Pagán et al. 2015). If the anomaly takes too much time to be removed, the `GPML` would not have enough recent data to estimate new values. In this case, when a predefined time is exceeded, the `ete<bool>` signal gets activated and the data from the sensor is not used for migraine prediction until a signal coming form the `Manual Status` indicates that the sensor has been restored. Then all alarms signals are disabled and `SSD` relays the data from the sensor without errors.

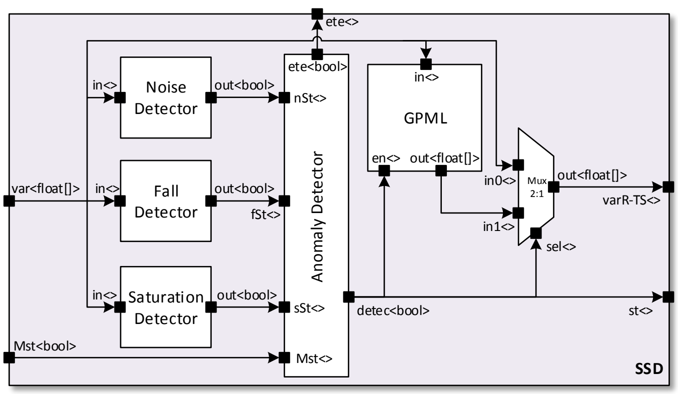
The `Predictor` (in Figure 3c) is the last type of coupled model and contains the migraine prediction models trained, and holds several atomic models. The `SDMS2` atomic model selects, subject to availability, the group of state-space models to perform the prediction. Each one of the atomic models `Predictors_*`—where '`*`' indicates the set of available sensors—computes several predictions (three in our case) and sends them to the atomic model `Linear Combiner`, which performs a linear combination of the three results. In our research we have developed state-space models and GE models for the migraine prediction; any of them, or others, can be used in the system. The `SDMS2` applies a hierarchy of models according to the availability of sensors to maintain prediction accuracy. In our future work, we aim to provide an advanced feature of the system that will allow the injection of re-trained migraine prediction models (`Predictors_*`) in real time. Then, the behavior of the system will be shown using variable structure DEVS (Hu et al. 2005).

3.2.2 Atomic models

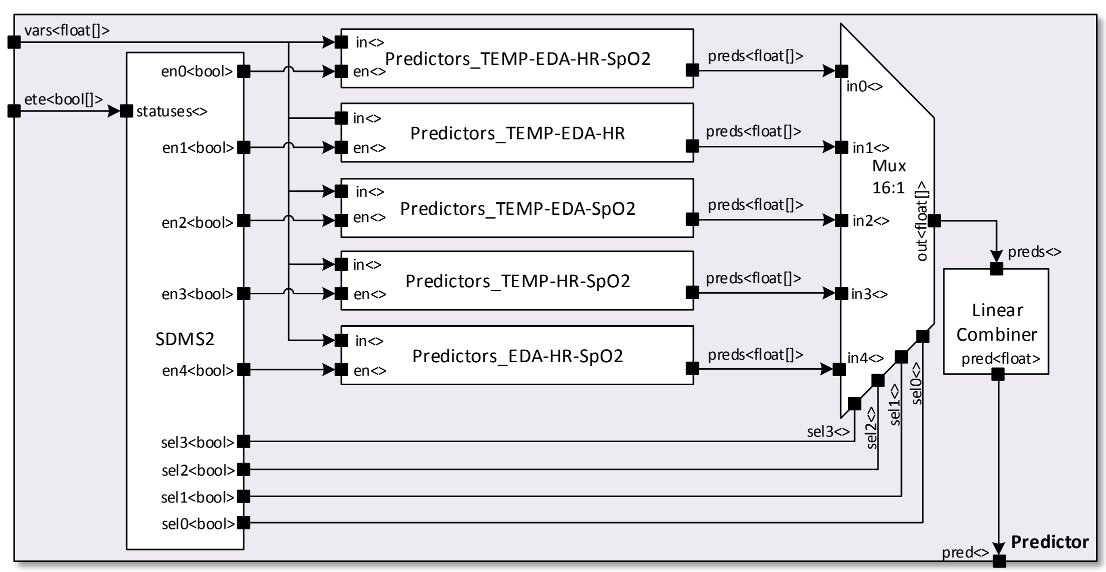
Among the twenty-five top-level atomic models, we distinguish seven different types: sensors and stimulus (`TEMP`, `EDA`, `HR`, `SpO2` and `Pain`), statistical error inductors (`ErrorInductor`), manual reset model for



(a) Coupled model `EFgt` that controls the data flow in the system.



(b) Coupled models `SSD` that check the status of sensors.



(c) Coupled model `Predictor` that computes the migraine predictions.

Figure 3: Three of the coupled models that the `EFsys` hosts.

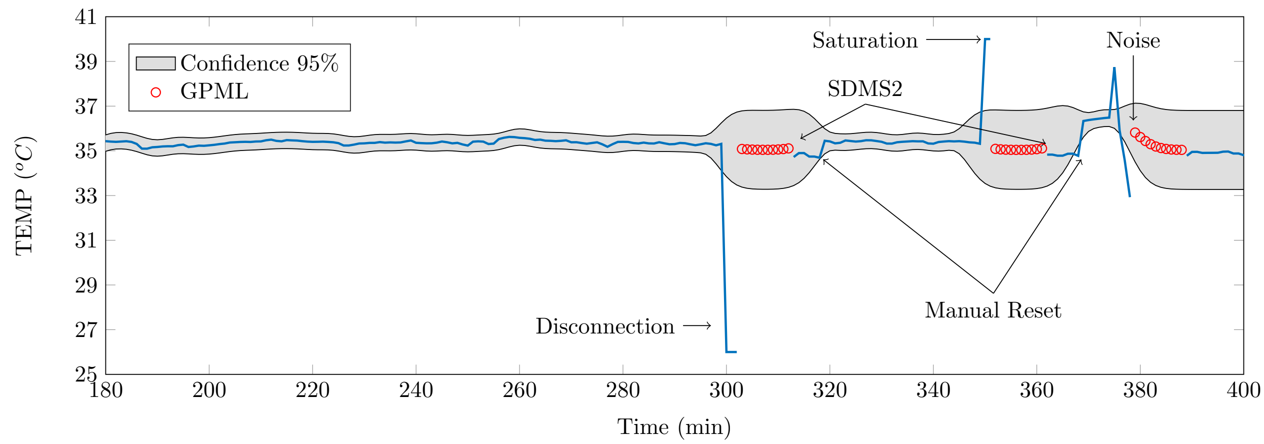
sensors (`Manual Status`), sensor drivers (`Sensor Driver`), signal synchronizer (`Sync`), an alarm evaluator (`Decider`) and the migraine alarm `Actuator`. All atomic models in the `EFsys` model represent HW modules that can be replaced by real HW components in a future HIL implementation.

- The atomic models: `TEMP`, `EDA`, `HR` and `SpO2` pre-process biometric variables in their current implementation. In the future release of the prediction system, these models will provide the raw data from sensors, and will also incorporate real-time processing models (in the `Sensor Driver` models after the `ErrorInductor` models). Atomic models for hemodynamic variables remain outside of the `EFsys` coupled model so that they can be substituted easily for hardware devices facilitating the execution of HIL experiments. `Pain` is the symptomatic pain curve modeled from pain mark levels indicated by patients as shown in (Pagán et al. 2015). This atomic model does not belong to the system and has been added to compute statistics when the framework is used with known migraine events.
- The error inductor modules `{TEMP-EDA-HR-SpO2} ErrorInductor` are atomic models that do not belong to the system but are useful in the simulation system to induce random errors. These models generate three types of errors: noise, saturation and disconnections (or falls). Errors are generated based on signals' error statistics, and these statistics are used in `SSD` coupled models to detect the errors.
- The driver `{TEMP-EDA-HR-SpO2} Sensor Driver` adds a timestamp to the data from the `RootCoupled`'s clock. In the current implementation, these models do not perform any action. In the future release, they might include the signal processing of raw data coming from sensors.
- The model `{TEMP-EDA-HR-SpO2} Manual Status` represents hardware that raises notifications when a damaged sensor has been repaired. The generated signal resets the alarms. This leads the `SDMS2` model to again select the set of prediction models using all the sensors. In an HIL implementation they will be replaced by buttons or something similar.
- The `Sync` atomic model synchronizes and buffers the data for simultaneously supplying the values for the four biometric variables (`Pain`, if possible) to the coupled model `EFgt`.
- The `Decider` is an atomic model that determines if prediction results in a migraine event or not. The `Decider` is implemented as a threshold crossing model with a single level only. The numerical threshold value is 32 (normalized units) in the normalized objective symptomatic pain curve and this represents 50% probability of the maximum pain level (Pagán et al. 2015).
- The `Actuator` is an atomic model that can be substituted by a hardware device, most likely an acoustic alarm. In the simulation system, this is a dummy model and it does not perform any action.
- The models `{TEMP-EDA-HR-SpO2} Status` indicate when sensors have data errors. In the simulation system, these are dummy models and they do not perform any action. They will be substitued by stimulus such as LED diodes.

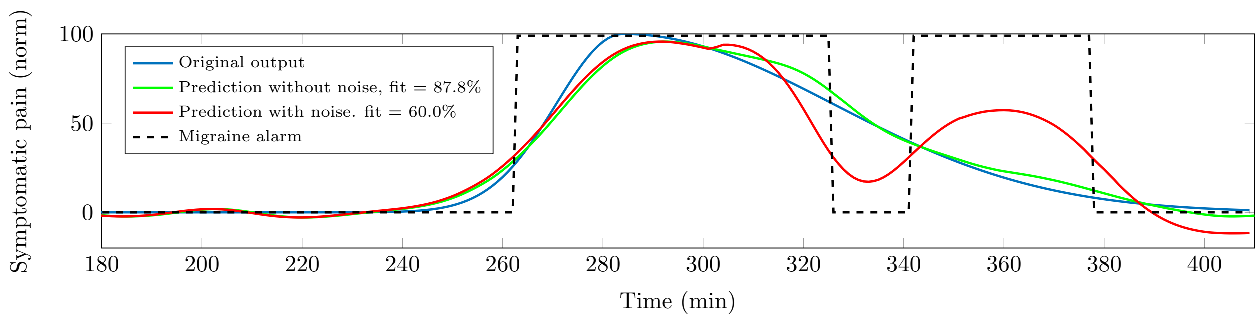
4 EVALUATION

In this section we describe the behavior of the system against sensor failures. Actual implemented migraine prediction models only support one sensor failure at a time. Migraine prediction using less that three sensors leads to low accurate predictions, which makes the system not useful (Pagán et al. 2015). When all sensors fail the system is not able to compute predictions and alarms will not be generated. All kind of failures can be tested, however, the complete sensor failure and the statistical study of accuracy and improvement as a result of that failure is out of the scope of this paper and will be presented in our future work. To demonstrate the fundamental concept, errors have been induced in only one sensor (temperature sensor).

In Figure 4a, three different types of errors have been randomly induced to the TEMP signal: disconnection, saturation and noise. In normal conditions, the `Predictors_TEMP-EDA-HR-SpO2` model in Figure 3c



(a) TEMP sensor and system events.



(b) System's output and alarm event.

Figure 4: Error induction in the temperature sensor and behavior of the system against the failures.

computes migraine predictions. When the `TEMP SSD` model detects an error, it activates the signal recovery through the `GPML` (red circles), and after a pre-specified duration (defined as 10 minutes in this example) the `SDMS2` switches the migraine predictor model to `Predictors_EDA-HR-SpO2`, which does not use the information from the TEMP sensor. The gray bands represent the 95% confidence level of the GPML, which is tight (better) when data are not missing and it is spread (worse) when the GPML algorithm recovers data. Despite being represented all along the signal, the gray bands really exist only when the `GPML` model works (red circles). In a real HIL implementation, when the TEMP sensor has been replaced or repaired, the `Predictors_TEMP-EDA-HR-SpO2` model will be used again. In the example shown in Figure 4a, two manual resets were simulated.

In Figure 4b, we can see how these errors affect the prediction (green curve). Recovering values through the prediction simulation system avoids destabilization of the migraine system despite an apparent drop in accuracy. In this example, the normalized root mean squared error (NRMSE) between the prediction with noise (red curve) and the prediction without noise (green curve) is only 12.2%.

With this simple example we have demonstrated how intermittent sensor failures can be replaced by predicted values to stabilize the entire system and keep it within an acceptable operating range. Certainly, a deeper evaluation is required in real-life conditions for ambulatory monitoring (not having to stay in bed). This would also required advanced behavior in the `Decider` model.

5 DISCUSSION

This paper describes an advanced approach for the development of a simulation system for a migraine prediction system. The migraine disease is one of the most disabling neurological diseases. The prediction of an incoming symptomatic event allows the patient to take painkillers in advance and thus stop the pain. Migraine prediction models have been developed in our previous work. A first step prior to the expensive and slow physical implementation of a complete prediction and monitoring device is to evaluate the system using modeling and simulation. Consequently, this paper provides an overview of the simulation of the migraine prediction system that raises an alarm at appropriate times and alerts the patient.

The simulator has been developed using the DEVS formalism and has been validated using state-space predictive models. This work allows testing the system's behavior under real conditions for ambulatory monitoring. The system has been demonstrated as a tool to raise alarms when sensor errors occur. It performs decision-making using the migraine predictive models to maintain the quality of the prediction. This work provides a proof-of-concept. The next step of this project will be an implementation in a portable device for an HIL system. This will then be followed by system design refinement leading to the final design of a migraine prediction device. This HW/SW co-simulation including HIL components is a useful method to evaluate failures in physical devices and the resulting consequences. Such approaches contribute to resilient cyber-physical systems engineering practices.

ACKNOWLEDGEMENTS

Research by Josué Pagán has been funded by the EU (FEDER) under Research grant PI15/01976 and the Spanish Ministry of Economy and Competitiveness under Research Grants TIN 2015-65277-R and TEC2012-33892.

REFERENCES

Barhak, J., D. J. Isaman, W. Ye, and D. Lee. 2010. "Chronic disease modeling and simulation software". *Journal of biomedical informatics* vol. 43 (5), pp. 791–799.

Barshan, B. e. a. 2014. "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units". *The Computer Journal* vol. 57 (11), pp. 1649–1667.

Buck, J. T., S. Ha, E. A. Lee, and D. G. Messerschmitt. 1994. "Ptolemy: A framework for simulating and prototyping heterogeneous systems".

Eddy, D. M., and L. Schlessinger. 2003. "Archimedes A trial-validated model of diabetes". *Diabetes care* vol. 26 (11), pp. 3093–3101.

Giffin, N., L. Ruggiero, R. Lipton, and et al. 2003. "Premonitory symptoms in migraine An electronic diary study". *Neurology* vol. 60 (6), pp. 935–940.

Hu, X., X. Hu, B. P. Zeigler, and S. Mittal. 2005. "Variable structure in DEVS component-based modeling and simulation". *Simulation* vol. 81 (2), pp. 91–102.

Linde, M., A. Gustavsson, L. Stovner, and et al. 2012. "The cost of headache disorders in Europe: the Eurolight project". *European journal of neurology* vol. 19 (5), pp. 703–711.

Lipton, R. B., and A. I. Scher. 2001. "Epidemiology and economic impact of migraine". *Current Medical Research and Opinion®* vol. 17 (S1), pp. s4–12.

Pagán, J., J. L. Risco-Martín, J. M. Moya, and J. L. Ayala. 2016. "A Real-Time Framework for a DEVS-based Migraine Prediction Simulator System". In *MAEB 2016*.

Pagán, J., , J. L. Risco-Martín, J. M. Moya, and J. L. Ayala. 2016. "Grammatical Evolutionary Techniques for Prompt Migraine Prediction". In *Genetic and Evolutionary Computation Conference, 2016*. ACM.

Pagán, J., M. I. D. Orbe, A. Gago, and et al. 2015. "Robust and Accurate Modeling Approaches for Migraine Per-Patient Prediction from Ambulatory Data". *Sensors* vol. 15, pp. 15419–15442.

Pagán, J., M. Zapater, and J. L. Ayala. 2017. "Power transmission and workload balancing policies in eHealth mobile cloud computing scenarios". *Future Generation Computer Systems*, pp. –.

Rasmussen, C. E., and C. K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Risco-Martín, J. L., M. Jesús, S. Mittal, and B. P. Zeigler. 2009. "eudevs: Executable uml with devs theory of modeling and simulation". *Simulation* vol. 85 (11-12), pp. 750–777.

Risco-Martín, J. L., S. Mittal, J. C. Fabero, P. Malagon, and J. L. Ayala. 2016. "Real-time hardware/software co-design using DEVS-based transparent M&S framework". In *SummerSim-SCSC, 2016*.

Stovner, L. J., and C. Andree. 2010. "Prevalence of headache in Europe: a review for the Eurolight project". *The journal of headache and pain* vol. 11 (4), pp. 289–299.

Wainer, G. A., R. Madhoun, and et al. 2008. "Distributed simulation of DEVS and Cell-DEVS models in CD++ using Web-Services". *Simulation Modelling Practice and Theory* vol. 16 (9), pp. 1266–1292.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press.

Zheng, Y.-L., X.-R. Ding, and et al. 2014. "Unobtrusive sensing and wearable devices for health informatics". *Biomedical Engineering, IEEE Transactions on* vol. 61 (5), pp. 1538–1554.

AUTHOR BIOGRAPHIES

JOSUÉ PAGÁN is a Ph.D. candidate at the Complutense University of Madrid (UCM) and works as a research assistant at the Technical University of Madrid (UPM). His work focuses on the development of robust methodologies for information acquisition in biophysical and critical scenarios, developing models for prompt prediction and classification of neurological diseases. His email address is jpagan@ucm.es.

JOSÉ M. MOYA is currently an Associate Professor in the Department of Electronic Engineering, Technical University of Madrid. He has participated in a large number of national research projects in the fields of design and optimization, and security of embedded and distributed systems. His research interests focus on proactive and reactive thermal-aware optimization of data centers, and design techniques and tools for energy-efficient compute-intensive embedded applications. His email address is josem@die.upm.es.

JOSÉ L. RISCO-MARTÍN received his Ph.D. from Complutense University of Madrid, and currently is Associate Professor in the Department of Computer Architecture and Automation at Complutense University of Madrid. His research interests include computer aided design, optimization and discrete event simulation. His email address is jlrisco@ucm.es.

JOSÉ L. AYALA is currently an Associate Professor in the Department of Computer Architecture and Automation at the Complutense University of Madrid. He has served as TPC member of many conferences, including DATE, DAC, VLSI-SoC, ICCAD, GLSVLSI, etc. His research interests focus on thermal and energy aware design and management of processor-based systems, design of embedded processors, thermal estimation, health monitoring and wireless sensor networks. His email address is jayala@ucm.es.