This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

Analysis of Smart Mobile Applications for Healthcare under Dynamic Context Changes

Ayan Banerjee* *Member, IEEE* and Sandeep K. S. Gupta* *Senior Member, IEEE.*

Abstract—Smart mobile medical computing systems (SMDCSes), e.g., mobile medical applications use context information from the environment to provide useful and often critical healthcare services such as continuous monitoring and control of blood glucose levels by infusion of insulin. Given the unsupervised nature of operation of SMDCSes, context changes that are unaccounted for can cause unprecedented faults leading to violation of requirements such as safety, energy sustainability and reliability. Analysis of SMDCSes for testing requirements violations necessitates consideration of context dependent interactions between the SMDCS software, represented by discrete operating modes and its environment, represented by non-linear partial differential equations over space and time. An intractable number of context change sequence and lack of closed form solutions to differential equations makes the requirements analysis of SMDCSes a challenging task. This paper proposes a novel technique to analyze SMDCSes taking into account the dynamic changes in the context and the constant interaction of the computing systems with the physical environment. To show the usage of the technique, Ayushman pervasive health monitoring system is considered as an example SMDCS. Analytical results show that practices considered healthy for a person such as mobility may not be beneficial when an SMDCS is controlling health.

Index Terms—Smart Mobile Applications, Pervasive Health Monitoring System, Safety, Sustainability, Medical Devices, Cyber-Physical Systems.

1 Introduction

Context awareness is a key feature of smart mobile medical computing systems (SMDCSes), that enable them to provide services with significant societal benefits such as mobile applications (apps) for healthcare. Application suites such as bHealthy [2] are becoming prevalent, where the smartphone uses a collection of applications to aggregate physiological and environmental data from sensors, store and process data to diagnose, display or control actuators, and communicate data to the cloud (Figure 1). Although context information is used to enhance user experiences or system performance, consequences of context changes can often be unwanted, such as loss of reception. In critical infrastructures such as healthcare systems, context changes may trigger hazardous consequences for the user. Indeed the Food and Drugs Administration (FDA) has considered SMDCSes in healthcare as medical electrical equipments and has recommended strict safety guidelines for them [3]. FDA has recognized three types of health apps: a) display apps, b) diagnostic apps, and c) controller apps and encourages the use of safety verification tools such as static software testing [4] for safety critical mobile apps. Further, a key feature of SMDCSes is their pervasive and unobtrusive operation. Hence, apart from safety a key requirement for SMDCSes is sustainability, i.e., their long term operation with limited energy from sources such as batteries or scavenging systems. This paper considers the safety and sustainability analysis of SMDCSes under dynamic context changes.

SMDCSes interact with the environment for gathering context information such as physiological or mental state of a person [2]. They may also be involved in administering critical actuation functions such as drug infusion. Hence, there is continuous interaction between the computing units and the physical environment through sensing, control, and actuation, referred to as *cyber-physical interactions*. Random context changes that are unaccounted for in the SMDCS design can cause uncontrolled cyber-physical interactions potentially risking the user's health [5]. A case in point is that of a wireless controlled analgesic infusion pump, where a controller sends control inputs to infusion pump over the wireless channel to maintain the analgesic drug concentration within a safe range. The controller gets feedback from a body worn pulse oximeter recording the current blood oxygen level. The pump should stop infusing immediately when the blood oxygen level falls below a certain level to prevent respiratory distress [6].

Since the wireless channel is prone to errors due to interference, the packets containing accurate blood oxygen level can get corrupted or dropped at random. If the controller does not obtain an accurate estimation of the blood oxygen level it can cause unstable or oscillatory infusion rates, which can induce respiratory distress. Thus, receiving the correct control information despite wireless interference is important. Wireless interference patterns change drastically with location context of the user e.g., the packet delivery rate of a medium

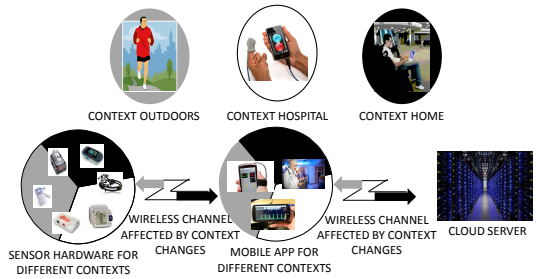

Fig. 1. **SMDCS system model consists of a suite of mobile apps each interfacing with different sensors and actuators. In each context (color coded) the user may use different mobile apps and sensor or actuator configurations.**

* *The authors are affiliated with School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Email:* {abanerj3,sandeep.gupta}@asu.edu. *This research was funded in part by NSF grants CNS-0831544 and IIS- 1116385, and gifts from Intel Corporation. Thanks to the OSEL group in FDA for providing infusion pump models and Intel for providing Atom platforms. A preliminary version of this paper appeared in IEEE Percom 2012 [1].*

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

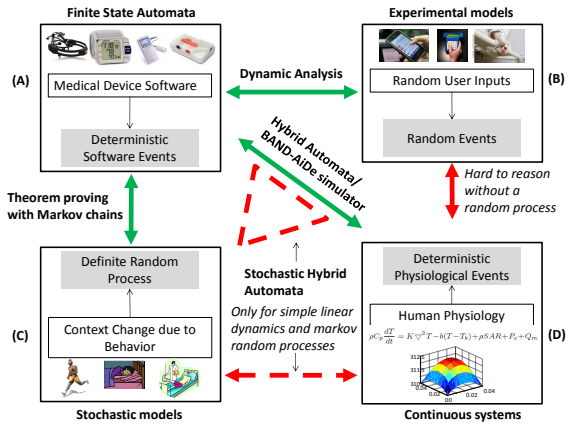

Fig. 2. **Gap analysis for existing techniques for unified evaluation of context, software, and physical processes. The solid black arrows show existing analysis techniques, which are also identified in bold text. The solid grey arrow indicates absence of techniques and the text in italics reasons why. The dashed arrows indicate that there are limited techniques available and the text in italics shows the reason why such techniques are not applicable.**

may vary [7] from indoors to outdoors. Location context changes are governed by user mobility. In such a scenario the user mobility pattern may be unsafe for his health [1]!

Since SMDCSes are intended for pervasive use and a faulty operation of SMDCSes may cause harm to human life, it is of utmost importance that their operation is verified against two principal properties - a) *safety*, i.e. avoidance of hazards to the user, and b) *sustainability*, long term operation using limited energy sources. Traditionally, SMDCSes are verified against such requirements using experiments in a controlled laboratory environment. Such methods are in-comprehensive since they ignore the dynamic context changes that occur in unsupervised environment where the SMDCSes are actually deployed. Further, the experiments are mostly performed after the implementation. Any fault detected in this phase may incur significant cost of re-implementation. Hence, it is beneficial to analyze safety and sustainability of SMDCSes before their implementation and deployment.

This paper provides a systematic non-invasive methodology for analyzing SMDCS properties under dynamic context changes and interaction of devices with the environment.

1.1 Challenges

To analyze an SMDCS four components have to be considered (Figure 2): a) the software of the mobile devices, which generates discrete events at deterministic times, b) random inputs from the user, which generate random discrete events at random times, c) dynamic context changes of the user, which change the user's environment following random processes and d) the human physiology, which change physiological parameters following a continuous dynamics.

Model based engineering techniques are being widely used as a non-invasive method for analyzing and verifying system design. In this paper, we propose a model based engineering (MBE) approach for analyzing safety and sustainability of SMDCSes. In this technique, before implementation and deployment of a system, a high level model is developed that mathematically characterizes salient features of the system with desired accuracy. Individual components of the model are then calibrated using real world experiments. The integrated model is then analyzed to evaluate the safety and sustainability properties of the system.

Existing MBE techniques can effectively analyze each of the SMDCS components individually, however, they are inadequate for analyzing the effects of dynamic context changes on the safety and sustainability of the whole SMDCS system. As shown in Figure 2, MBE techniques has been used individually for the four components of SMDCS.

a) Medical device software and hardware: A large number of tools are available that model and analyze hardware of computing systems such as Pspice [8] and Architectural Analysis and Description Language (AADL) (http://www.aadl.info/), and application software such as Unified Modeling Language (UML) (http://www.uml.org/) and Petrinets [9]. The *ANDES* tool [10] uses MBE in Wireless Sensor Networks (WSNs) to ensure accuracy and low latency of WSN operations. Finite state automata (FSA) and timed automata can be used to theoretically analyze safety or sustainability of medical device software [3], [11], [12]. Further, static analysis techniques [4], [13] can be used to check the correctness of code.

b) Random user inputs: The random user behavior with the mobile devices are typically represented using empirical models such as exponential and poisson processes.

c) Context dependent human behavior: Human behavior, governed by their day to day activities, is random with some form of periodicity. Common models used to represent user mobility are random walks and Markov chains [14].

d) Continuous physical systems: The MBE approach is also used to study the behavior of physical systems through tools such as SysML (http://www.sysml.org/), Simulink (http://www.mathworks.com/), and Flovent (http://www.mentor.com/). The human physiology is mostly represented using deterministic differential equation models and hence can be analyzed using well established continuous system theories.

The analysis of SMDCSes on the contrary necessitates integrated analysis of different components. In such cases, techniques such as hybrid automata [15]–[17], dynamic analysis [18], and stochastic processes [19] are applied. In a hybrid automata the software behavior is modeled using a FSA and the system variables, which represent physiological condition of a user, are governed by deterministic differential equations. The transitions between different discrete states in a hybrid automata are governed by deterministic events when system variables cross certain pre-specified thresholds. Hybrid automata is not applied to model random context changes.

Dynamic analysis is a widely used technique to test operation of software under random user inputs. However, it does not incorporate interaction of the software with the human physiology. Stochastic hybrid automata [19] is an advanced tool that can handle interaction of software events with human physiology and also allow randomness in state transitions. However, the analysis is only limited to linear ordinary differential equations and mostly exponential or Poisson random processes, which have finite variance. Apart

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

TABLE 1
Classification of Existing Work on Model Based Analysis of Medical Devices.

Class	Approaches	Device Modeling	Physical Modeling	Interaction Modeling	Domain	Dynamic Contexts
Hardware/ Software Modeling	PSpice [8], VHDL and Verilog [20], AADL [21], UML [22], ANDES [10], Hardware Testbeds [23], [24]	✓	✗	✗	Multiple	✗
Physical Modeling	MATLAB ® and Simulink [25], SysML [26], Flovent [27], TrueTime [28], Modelica [17], Theoretical modeling of human body such as heart models [11], diffusion models [29]	✗	✓	✗	Multiple	✗
Interfacing tools	BAND-AiDe [30] (AADL + MATLAB ®), LabView + Ptolemy [15], MATLAB ® + DeterLab [31], Multi-domain modeling using architectural views [32]	✓	✓	Limited capability	Mostly Single physical domain with the exception of Ptolemy [15] and multi-domain modeling [32]	✗
Integrated Modeling	Modelica [33], BAND-AiDe [30], Hybrid Quartz [34]	✓	✓	✓	Limited in representing computing systems	✗
Formal Modeling	Timed Automata [3], [11], [12], Petrinets [35], Hybrid Systems [15]–[17]	✓	✓	✓	Multiple	✗

from these theoretical tools, there are several modeling and simulation tools available for analyzing an SMDCS for satisfaction of system requirements, referred to as Requirements Analysis henceforth. However, they fail to comprehensively analyze SMDCSes for safety and sustainability under random context changes and non-linear spatio-temporal interaction. A complete list of such tools is provided in Table 1.

The *main research problem is to analyze the interaction between software, user context, and human physiology*, which is complicated by several aspects:

a) Spatio-temporal dynamics exist in human physiology, which often do not have closed form solutions hence making theoretical proofs difficult. Recent research has proposed Spatio-Temporal Hybrid Automata (STHA) [36] to model and analyze linear spatio-temporal dynamics and software events in a single framework. However, they fail to consider stochastic nature of context changes and user inputs.

b) Aggregate effects occur when multiple mobile devices interact with the same user to control its physiology, e.g., multichannel infusion pumps infusing both glucagon and insulin to control blood glucose levels. The effect of simultaneous administration of the two drugs is significantly different from isolated administration. These aggregate effects can occur anytime and anywhere the two drugs interact and have to be approximated using intricate models [36]. The BAND-Aide [30] tool can simulate aggregate effects for only linear systems and it does not consider evolution of aggregate effects under context changes.

c) Nonlinearity in physiology requires more intricate analysis. Although several tools such as KeyMaeraD [37] have been proposed but they incur errors due to piecewise linear or rectangular approximations.

d) Heavy tailed random processes often arise in practice (Levy walk mobility patterns) and hence can have infinite variance [14]. In such a scenario, analyzing the effect of dynamic context changes on spatio-temporal interaction becomes a challenging task.

e) Potentially infinite number of context sequences have to be analyzed to guarantee requirements of an SMDCS model. Further, even if the context sequences are limited to a length n, a case by case context analysis procedure can take exponential amount of time.

To avoid such computational complexity, this paper takes an integrated specification and simulation analysis approach, where contexts and physical processes are specified in the same framework. The paper then proposes safety and sustainability analysis algorithms of polynomial complexity that can be used to simulate the execution of a SMDCS model with spatio-temporal aggregate interactions for a set of contexts.

1.2 Contributions

Overall, the main contributions of this paper are:

- development of an integrated specification logic for SMDCSes, which enables specification of dynamic user context changes and the cooperation of the computing system with the user environment;
- integrating models of computation and physical system to develop polynomial time randomized analysis algorithm for requirements analysis;
- probabilistic runtime estimation of the analysis algorithm;
- comprehensive case studies showing the usage of the proposed methodology on Ayushman SMDCS and experimental validation of the design in a hospital setting.

This paper considers Ayushman, a pervasive health monitoring system as an example SMDCS to demonstrate the usage of the model based approach proposed in the paper. Using the infusion pump example we show how the mobility pattern of a user can harm the drug diffusion safety. Energy sustainability analysis shows how context triggered health emergency detection algorithms can deplete energy sources faster. Finally, we show how the mobility of an user can beneficially affect the sustainability of the SMDCS. We use industry standard AADL (www.aadl.info) to implement the specification and analysis phase. AADL allows extensions by introducing new language constructs as *annex*. We next discuss

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

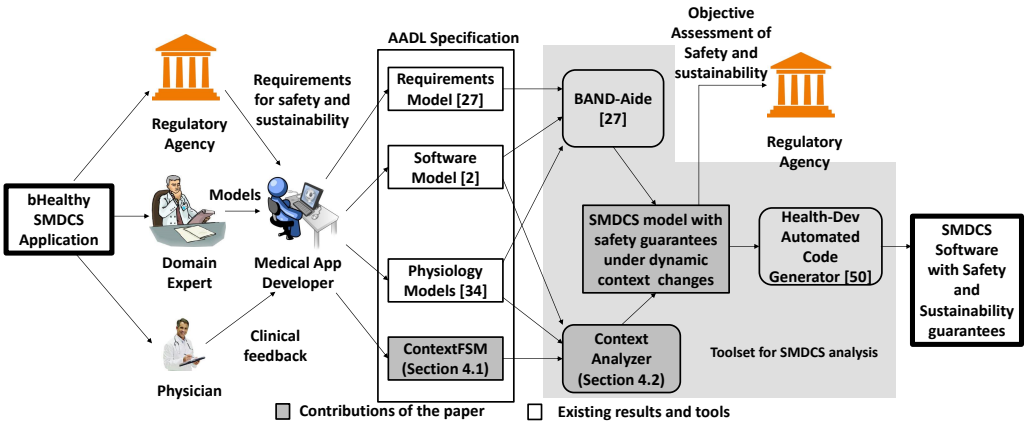

Fig. 3. **An app developer obtains information from regulatory agencies, domain experts, and physicians and uses the proposed approach along with BAND-Aide, and Health-Dev tools to analyze and implement safety assured software for the bHealthy application.**

the usage of the proposed context analysis methodology in mobile app development.

1.3 App development using the proposed framework

The context analysis algorithm proposed in this paper is a part of a comprehensive safe and sustainable mobile app development methodology that requires collaboration between the medical app developer, regulatory agencies, domain experts and medical practitioners. In a typical use case of our proposed methodology (Figure 3), a developer may consider developing a mobile application such as bHealthy [2]. It is a collection of physiological feedback-based mobile applications to assess the mental state of a user, suggest activities that promote user well-being, and compile a wellness report. The developer designs the application specification and consults safety (or sustainability) guidances provided by regulatory agencies. The developer then employs a team of domain experts to develop models for different components of the SMDCS. For medical applications consultation with physician is also required to translate the regulatory requirements to design constraints. The developer can then input these models to the context analyzer and other external tools to perform safety and sustainability analysis. High level modeling language such as AADL can be used for this purpose.

For analyzing the spatio-temporal interactions, without context changes, the developer can use BAND-AiDe [30]. The AADL models of spatio-temporal interactions can be converted to hybrid automata to perform more formal analysis. To verify the safety and/or sustainability of the SMDCS models under dynamic context changes, the developer can use the context analyzer proposed in this paper. The analysis results in SMDCS models with safety and sustainability properties. These models can be provided as supporting documents for a market approval process to a regulatory agency. Further, these models can be converted to implementations in sensors and smartphones either manually or through an automation software. In our previous work, we have proposed Health-Dev [38], an automatic code generator, that takes AADL models as input and converts them into implementations. Automated code generation minimizes human errors and esnures that implementations have the same properties as the models.

2 System Model

In our system model we consider context to be a fixed evaluation of variables of a system. For example, as shown in Figure 1, the home context may have a fixed ambient temperature of 37 °C, the user may use an electrocardiogram (ECG) sensor at 250 Hz sampling frequency, which stores data in a smartphone using a specific mobile application. The core of an SMDCS is a set of mobile applications, intended to be used in different contexts, that collect data from sensors, display them to the user through a graphical interface, or process them or provide some form of bio-feedback. Note that some sensors and mobile applications can be used in multiple contexts. The data may also be communicated to a cloud service which is used as a storage or computation hub. An example SMDCS is the bHealthy application suite developed at the IMPACT Lab [2]. In this application suite there are two applications *PetPeeves* and *BrainHealth* that are intended to be used in two different contexts. PetPeeves uses accelerometers and ECG sensors to measure exercise levels and calories burnt and provides a bio-feedback to the user through animations of a virtual pet. It is intended to be used outdoors while exercising. BrainHealth, an application to be used at home, employs electroencephalogram (EEG) and ECG sensors to derive the users concentration levels and engages them in a neurofeedback based game to increase their concentration.

3 Effects of Context Changes

Context is a set of information that can characterize the state of a computing system or the human body [39]. The set of information may include physiological condition, mood, and time of the day. Each context affects the human body parameters in different ways. For example, during hot and humid summers the body sweats leading to a lower average skin conductance, or when a person is excited the skin conductance increases. Since these parameters affect the way a medical device interacts with the human, a change in context leads to

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

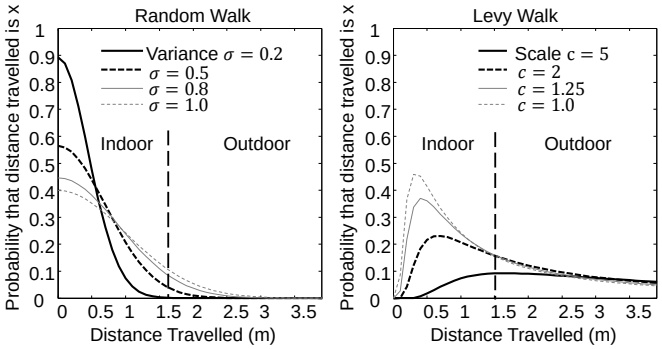

Fig. 4. **Longer distances are less likely to be traveled in Random way point than in Levy walk mobility models.**

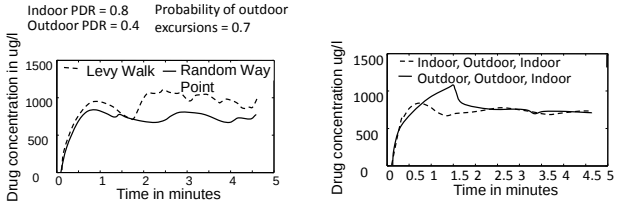

Fig. 5. **a) Drug concentration have different overshoots for different mobility patterns, b) drug concentration profile may depend on the sequence of context changes.**

a change in medical device performance. Mobility is a basic human nature, which lead to context changes affecting the interaction between a medical device and the human body. For example, consider a wearable autonomous infusion pump.

Infusion Pump Control System: The infusion pump [6] is a medical electrical equipment that obtains commands from a remote computer or a smart phone over the wireless channel and accordingly injects a dose of drug such as insulin or anaesthetic into the human body. The controller obtains feedback from the human body using sensors such as a glucosemeter and according to a control algorithm computes the future infusion rate and sends it to the pump. In the literature, the feedback has been modeled by pharmacokinetic diffusion equations [6], which take the infusion rate as input and outputs the drug concentration in the blood. The controller then calculates the infusion rate so that the drug concentration is maintained within a prescribed range without overshooting. If the estimated drug concentration goes beyond the desired range, the controller reduces infusion rate such that the estimated drug concentration remains within the range.

An important factor in this infusion pump device is the transfer of information from the controller to the pump through the wireless channel. Wireless channels are prone to errors leading to loss of control information. When the pump fails to receive a control information packet the pump maintains the infusion rate obtained in the last successfully received information [12]. Mobility affects the wireless channel characteristics leading to time varying packet delivery ratio (PDR), which may affect the drug concentration due to loss of infusion information. These effects further, vary with different mobility patterns. Hence to characterize these effects different models of mobility have to be studied.

Mobility Models: Over the years several models of human mobility have been studied including the random walk and Brownian motion models [14]. The most popularly used mobility model is the random walk model. However, recently, the Levy walk mobility model was found to fit the average human mobility the best [14]. A mobility model consists of three parameters: a) flight length, which is the distance traveled, b) flight direction, direction of the movement, and c) pause time, time for which the person stays at a particular position. The random walk mobility model assumes that the probability of flight length being greater than a certain value follows a Gaussian distribution. Thus, it is less probable for a person to move further away from a given spatial location. However, a recent research has shown that the flight length for human mobility follows a power law distribution or Levy distribution. This comes from the ever inquisitive nature of human being, which compels her to explore remote regions [14]. Instances of the two mobility models are shown in Figure 4, where in random walk the shorter flight lengths are more frequent, while Levy walk model has more frequent longer flight lengths.

Effect of mobility patterns on infusion pump operation: The infusion pump control system was simulated under different mobility patterns of the user. Two different channel properties were considered: a) indoor, with a PDR of 0.8 and b) outdoor, with a PDR of 0.4 as suggested in [7]. We took a stretch of 20 feet with a door separating indoor and outdoor environment at the 10 feet mark and computed the sequence of indoor and outdoor movements for random and Levy walk models. Packet drops were simulated using the Ricean fading model and the average case drug concentration for 1000 runs for both the mobility models is shown in Figure 5a). The figure shows that since the Levy walk mobility pattern has more frequent outdoor visits, it causes more loss of control information and hence causes drug overshoots due to faulty infusion. On the other hand random walk has shorter excursions leading to less frequent change of environment and hence less overshoot. Further, apart from the frequency of outdoor visits, different sequences of indoor to outdoor transitions also affect the drug concentration as shown in Figure 5b for Levy walk model.

4 Context Analysis Algorithm

In this section, we define some basic concepts that lead to a context analysis algorithm and its runtime estimation.

Definition 1: **Context:** Formally a context is defined as a tuple $\{G, M, I\}$ such that G is a set of system variables, the attribute set M is a set of real numbers, integers and strings, and I is a bipartite graph mapping between the sets of variables G and attributes M.

As an example, let us consider that the user of an SMDCS is at home. The set of variables can include the PDR quantifying the wireless channel characteristics, the SMDCS device characteristics such as infusion rates, glucose meter sensing frequency, smartphone data communication rate, blood pressure sensor sensitivity, and the human physiological parameters such as skin resistance. Each element in the attribute set M consists of a real number (infusion rate = 325.6 μg/dl) or integer (sampling frequency = 250 Hz) or a string (insulin

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

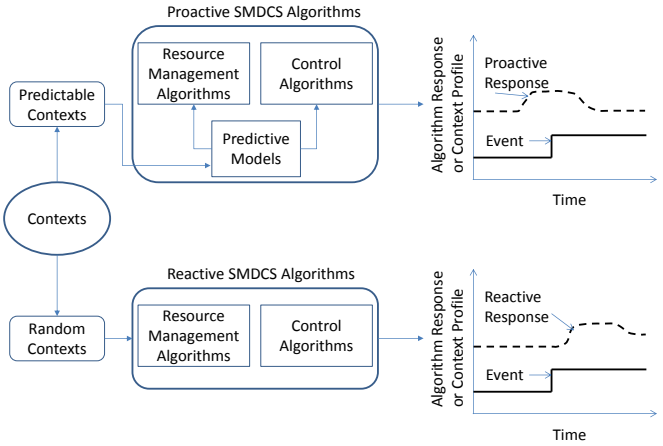

Fig. 6. **Contexts in SMDCSes are used for both reactive and proactive decision making.**

glucose interaction follows Bergman's minimal model [29]) corresponding to each element in the variable set G. Note that this way of representation can also be used to associate a physiological model with a physiological parameter.

A context change can be represented by a change in the object set G, attribute set M, and bipartite mapping I. These changes occur due to random processes in the environment. In this work, we consider three causes of context changes:

a) Mobility: An user of SMDCS is often in a state of motion in her daily life. This leads to frequent change in environmental properties such as temperature and humidity, or wireless channel properties such as the packet delivery ratio (PDR) of indoor and outdoor environment or location. This change is exhibited by a change in the mapping I where the same set of variables, humidity, temperature, PDR are mapped to different values.

b) Physiology: Random physiological events such as epileptic seizure can cause changes in the SMDCS or in its operation. It can introduce new medical devices such as a Holter monitor in a hospital, or it can cause execution of a new algorithm for analyzing specific disorders such as epilepsy. Introduction of the new sensor changes the object set G and subsequently the sets M and I.

c) User activities: Random user activities such as exercise or food intake can cause changes in the SMDCS. For example, during exercise the energy scavenged from the scavenging source may be sufficient for sustaining the operation of the computing units. This is exhibited in a change in the attribute set M of the SMDCS, in specific the scavenged energy attribute of the source.

The causes of the context changes are random in nature and hence have to be modeled using random processes. For example, human mobility is generally modeled using random processes such as Random way point or the Levy walk model.

Contexts in SMDCSes are used in two different ways (Figure 6): a) pro-actively involve context in the SMDCS operation, and b) react to the changes in the context. In *proactive* SMDCSes, the computing system uses predictive models of the context and incorporates them into decision making. Examples include model predictive wearable infusion pump algorithms, where physiological context of the user is predicted using pharmacokinetic models that express drug diffusion in the human body. The pharmacokinetic model is then used to estimate future blood glucose level excursions for the current infusion action. If the estimations cross thresholds then the control system changes mode to compensate for improper excursions in the future. Hence, the SMDCS is proactive in preparing for future contexts.

In *reactive* SMDCSes, the computing system does not estimate future contexts, but if a context change occurs it has algorithms and mechanisms to react and adapt. Examples include location aware mobile computing applications which give suggestions for good food places nearby. In such systems, a mapping of the desired output for each context is maintained. When a context change is detected using sensors or through fusion of information from various sources, the appropriate context to output mapping is used. Thus, although the SMDCS design is not aware of the context change the system reacts to it. In both the cases, contexts can be represented using some organization of discrete states and transitions between them.

4.1 Mathematical representation of user behavior

User behavior can be represented mathematically as a combination of contexts and probabilistic models of context changing events. Contexts and context changing events can be combined in a mathematical construct similar to a finite state machines, called *ContextFSM*.

Definition 2: **ContextFSM:** A ContextFSM is a tuple $\{X, T\}$, where -

- X is set of states. Each state corresponds to a context, i.e. an instance of the tuple $\{G, M, I\}$.
- T is a transition matrix, where $p_{i,j} \in T$ is the probability that there is a transition to state i from state j.

Associated with every ContextFSM is the notion of continuous time t over which the states in the set X evolve. Further, we note that although time continuously evolves in a ContextFSM the state changes are always discrete. Thus, we represent the t^{th} state in the ContextFSM execution by $X_t = x_i | x_i \in X$. The state transitions are governed by the context changing events. The events are generated at random and are governed by the dynamics of an underlying random process. The parameters of the random process are dependent on the user's mobility, physiology and activity patterns. Markov random processes are quite common in nature. For example, the mobility of human being is strictly Markovian in nature [14]. The Markovian assumption entails that the ContextFSM will have a memoryless property. We denote $P(X_t = x_i)$ as the probability that the t^{th} state in the ContextFSM execution is x_i. Then, according to the Markov property, the probability that the t^{th} state is x_i only depends on the $(t-1)^{th}$ state in the execution sequence of the ContextFSM, i.e., $P(X_t = x_i | X_{t-1} = x_j, X_{t-2} = x_k \ldots) = P(X_t = x_i | X_{t-1} = x_j)$. This memoryless property is only exhibited if the time t_i spent by the ContextFSM at a particular state x_i, follows an exponential distribution.

The transition probability matrix is determined from the probabilistic models of the context changing events. For example, if we consider the Levy walk mobility model of SMDCS user, then the probability of going outdoors follows

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

an inverse power law distribution [14]. The parameters of the model can be obtained by accurately calibrating against experimental data. As shown in Example 1 the occurrence of epilepsy is a context changing event and follows a Poisson process with an average value of 0.12 times per day obtained through calibration [40].

4.2 Context analysis methodology

From Definition 2 each state in the ContextFSM corresponds to a context $\{G, M, I\}$. For each context we have to analyze the interaction between software and the human physiology. The interaction has two parts: a control algorithm CA and an interaction function CPF. The CA is specified as a sequence of finite number of steps each with a deterministic result. The CPF is specified as a differential equation, which can be time delayed, non-linear, and even multi-dimensional partial differential equations. In one or more of the steps of CA the CPF has to be solved. We define an *execution* of cyber-physical interaction in Definition 3.

Definition 3: An *execution* of cyber-physical interaction for a given time t is the deterministic evaluation of each step of the control algorithm CA to determine control outputs (CO) and solution of CPF at designated steps of the CA.

To analyze context changes and their effects on SMDCSes we define a simulation of the ContextFSM in Definition 4.

Definition 4: A simulation of ContextFSM is a sequence of *executions* of cyber-physical interactions such that -

- each execution corresponds to a unique context $\{G_i, M_i, I_i\}$,
- two executions do not have the same context, i.e. $I_j \neq I_i$ for $i \neq j$.

Ideally a comprehensive analysis of SMDCS under dynamic context changes will require a simulation that cover all possible context change sequences. However, the total number of possible context change sequences is infinite and hence the simulation will run for an infinitely long time. Instead we only consider a simulation, which covers all possible sequences of length n, which is a user defined parameter. Even covering all possible context change sequences of length n or less will require exponential runtime of the simulation. To convert this exponential runtime of SMDCS analysis under dynamic contexts to a manageable polynomial runtime we considered a randomized simulation of ContextFSM.

To simulate a ContextFSM, we need two processes: a) an event generation engine, which runs the underlying Markov chain and obtains a sequence of states, and b) the analysis execution engine (AEE), that evaluates the control algorithm and solves the partial differential equation. The event generation engine has the following steps -

- Choose an initial state x_0 from the pool of states X.
- Determine the time for which the ContextFSM is in state x_0 by random sampling from an exponential distribution.
- Choose a number within the range [0,1] from a uniform distribution.
- If the number is greater than p_{x_0,x_j} go to the state x_j and repeat the procedure.
- Else stay in x_0 and repeat the procedure.

The analysis execution engine consists of evaluating the control algorithm and solving the partial differential equations expressing interaction of software with human body. The context analysis Algorithm 1 takes the SMDCS models, context models, the control algorithm and the interaction function as input and outputs a map of the system parameters over space and time, called *interaction map*. This CPF is typically

Algorithm 1 Interaction Map **IM** = **AEE**(Context Model **CM**, **SMDCS**, Time **T**, Space **S**, Control Algorithm **CA**, Interaction Function **CPF**, Threshold Probability P_{th}, Sequence Length n)

```
Event Queue = Simulate CM(T,S,P_th,n)
Initialize interaction map IM
while Event Queue ≠ empty do
    Next Event = POP(Event Queue)
    Current Context CC = Simulate ContextFSM for the Next Event.
    Current SMDCS Model CSM = GetModel(SMDCSes, Next Event Type)
    for t=0:τ: Time Before Next Event do
        Control outputs CO = execute control algorithm CA(CSM,IM)
        Update inputs to the CPF according to CO
        IM = Simulate interactions using CPF(τ,S)
    end for
end while
Check compliance with requirements expressed as thresholds on the IM
```

a solver for the partial differential equations expressing the human physiology. The AEE simulates the context model and generates events in an event queue. It then initializes the interaction map with initial conditions supplied as a part of the SMDCS model. The AEE then processes events from the event queue and causes transitions in the ContextFSM. Upon each transition to a new state, the AEE parses the new SMDCS model, processes the control algorithm to determine control outputs, and continues computing the interaction map using the function CPF by incrementing time in steps of τ until the time for the next event. This interaction map is then compared with threshold based requirements to test compliance. The time interval τ is selected depending on several factors two of which concern with the convergence of the solution of the partial differential equation expressing the interaction and the length and type of context sequences that need to be simulated.

4.3 Runtime of proposed context analysis algorithm

The runtime of the proposed context analysis algorithm depends on two aspects: a) the number of steps for which the event generator is run, and b) the time taken by the AEE, to compute CA and CPF. Note that the AEE is always run for a finite simulation time, which is equal to the inter-event time. For a given length of context change sequence, the event generator can be run for exponential amount of time to generate all sequences of length n. Essentially, the ContextFSM is a Markov chain, such that the time spent in a state is exponentially distributed and the transition probabilities are obtained from models of context changing events such as mobility. The rich literature of hitting time analysis of Markov chains presents us with a theory that can be used to perform a randomized analysis of our context analysis algorithm. This randomized run is much faster and can cover important context change sequences, which can cause safety violations.

The hitting time for a state x_i in a ContextFSM is the time at which the state x_i is first observed by the event generation

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

engine. Similarly, we can define the hitting time of a context change sequence, as the time at which a context change sequence, $x_1 \dots x_j$ is first observed by the event generation engine. Let us select a context sequence $x_0, x_1, x_2 \dots x_n$. Thus, the time for which the context analysis algorithm has to be run to obtain a sequence B is obtained from Theorem 4.1.

Theorem 4.1: The expected time, τ_B, that the sequence B of length n is first observed in a simulation of the ContextFSM is given by the moment generating function [41]-

$$E[z^{\tau_B}] = \frac{1}{1 + (1-z)B * B(z)}, \quad (1)$$

$$\text{where, } B * B(z) = \sum_{1 \le j \le n} \{ \frac{z^{-j}}{P(X = x_1) \dots P(X = x_j)} \}.$$

The $*$ operator is a convolution in the discrete z-transform domain. Theorem proof is in the online Appendix [42].

The runtime time τ_B of the algorithm can be computed by taking the first derivative of $E[z^{\tau_B}]$ with respect to z and setting $z = 0$. The probability of a context change sequence of length n can be easily determined as $P(B) = P(X = x_1)P(X = x_2) \dots P(X = x_n)$. The context analysis algorithm can be provided with a sequence B of length n with a given probability $P(B)$. The algorithm can then be run for τ_B amount of time so that sequences with probabilities of occurrence greater than $P(B)$ can be simulated.

For a ContextFSM that has only two states x_1 and x_2, the time τ_B is given by $\tau_B = \frac{(1-p_{12}^n)}{(1-p_{12})p_{12}^n}$, where p_{12} is the transition probability from x_1 to x_2. For values of $p_{12} > 0.5$, and close to 1, τ_B is $O(n^2)$. If we set high probability thresholds, Algorithm 1 can be run for polynomial time to simulate sequences of greater probability of occurrence than the threshold. Thus, ContextFSM can be simulated for τ_B time to generate all sequences of length $\le n$ and probability $\ge P_{th}$.

5 Implementation

In this section, we discuss a specification framework and automating the context analysis Algorithm 1.

5.1 Specification of SMDCS models

The specification of an SMDCS is done using the industry standard AADL language (www.aadl.info). AADL is a hierarchical model specification tool that provides constructs dedicated to modeling embedded software and hardware. However, AADL inherently does not support specification of context and context transitions and physical dynamics of the human body. We use the behavioral annex to specify the *ContextFSM*. Further, we extend AADL to incorporate specification of complex physical processes as a series of differential equations through the development of an annex.

SMDCS specification: In an SMDCS for each context, there is a different hardware and software implementation indicated by the subcomponents in the AADL Spec 2 (in the online Appendix [42]), and are specified using the **system implementation** construct. Each context is specified using the **mode** construct, and the context changes using mode transitions. The events are specified using the **features** construct. The events are generated from the context models specified in the *ContextSensor* **system** component. In Ayushman, there are *four* contexts - home, roaming, hospital, and inactive, and six events - RoamingActive, AtHome, Emergency, Mitigate, Activate and DeActivate. In each context, the SMDCS consists of context sensor nodes, energy source, and the coordination between devices and human body. In this paper, we discuss context and interaction specification, other components are in the online Appendix [42].

Specification of coordinated operation: The coordinated operation results in changes in the complex physical processes with events occurring in the computing domain. In the infusion pump example, the diffusion of drug is governed by the pharmacokinetic (PCK) process [6], which can be modeled as a spatio-temporal differential equation. However, the equations change with the change in state of the controller. The controller algorithm takes the drug concentration predicted by the PCK process as input and varies the infusion rate to keep the drug concentration at a given level. Such an algorithm can be represented using a state machine, which captures both the computing and physical behavior of the infusion pump. A hybrid automata can be used in this regard to capture the continuous physical dynamics in each state. However, the **mode** construct cannot be used since there is no provision to specify equations for a given state and transitions cannot depend on the variation of a system variable. Instead we use a combination of the *behavior_annex* and the *CPS_annex* in AADL [30] to specify the control algorithm as a hybrid automata, AADL Spec 4 (*NetworkControlledDevice.InfusionPump*, specification provided in the online Appendix [42]). We can specify the partial differential equations using *CPS_annex*, *PDE*1 and *PDE*2 and associate them with states *s*1 and *s*2 in the *behavior_annex*. Further, the events in the *behavior_annex* can occur when a variable in the implementation goes over threshold (*Overshoot* event).

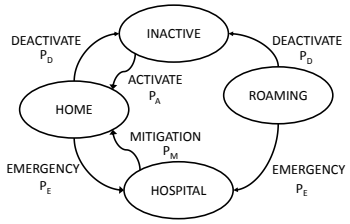

Fig. 7. **Finite State Automata representation of contexts and context changes *ContextFSM*.**

5.2 Context analysis algorithm implementation

The implementation of the context analysis algorithm (Algorithm 1) for requirements verification is shown in Figure 8. The first step in the analysis procedure is to generate context transition events. In this step, a random sequence of events are generated according to the ContextFSM and random processes characterizing the human behavior such as mobility models in case of mobile communication, arrhythmia occurrence probability in ECG monitoring, and bolus request frequency in infusion pumps. The random process takes the probabilities from the transition matrix T of a ContextFSM

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

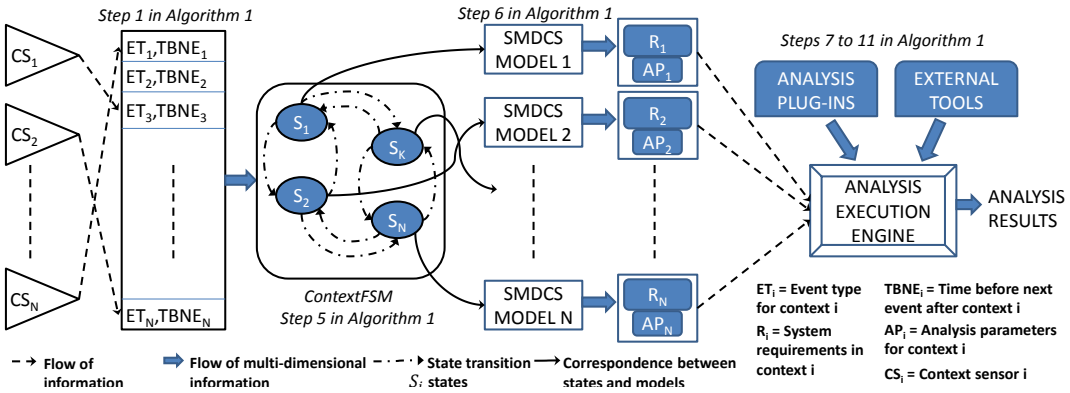

Fig. 8. **SMDCS analysis implementation- Event queue maintains triggers that change context, each context has a corresponding SMDCS model, which are analyzed by the analysis execution engine.**

and generates the context changing events. These events are classified into event types (ET_i) and are appended with an estimate of the time before next event ($TBNE_i$) and arranged into an event queue. The *ContextFSM* is then simulated starting from the initial state in accordance with the events. Each state maintains an SMDCS model specific for the context it represents. In each state, the context specific SMDCS is parsed to obtain the requirements and analysis parameters. Depending upon the requirements different analysis plug-ins are employed to perform the simulation of the SMDCS model. Further, domain specific tools such as MATLAB ® can also be used to analyze the SMDCS model. The execution of the appropriate plug-in for the correct analysis parameters and checking the compliance with the requirements is performed by the analysis execution unit (AEE) (Figure 8). The output of the AEE is the variation of the system parameters over time and space, *interaction map* (details in online Appendix [42]).

6 AYUSHMAN SMDCS

Ayushman [43] is a smart health infrastructure developed in the IMPACT Lab for privacy ensured continuous health monitoring of ambulatory individuals. It has a multi-tier architecture enabling management of sensors, secure storage and dissemination of data, access control of user health history, query processing, service discovery and context processing. At its core is a body sensor network (BSN) [44] consisting of a number of physiological as well as environmental sensors such as photo-plethysmogram, electrocardiogram, temperature, and humidity sensors and a smart phone serving as the computation and communication hub. On the smartphone end Ayushman uses bHealthy application suite to collect, store and process data. bHealthy has two applications: a) BrainHealth, which assesses the mental state of a user from EEG sensors and engages the user in a neurofeedback based game to increase focus levels, and b) PetPeeves, which uses accelerometer and ECG sensors to compute calories burnt and motivates the user to exercise by providing visual feedback through virtual pet. In Ayushman we consider three different contexts (Table 2), which vary in hardware software configurations, communication protocols, and power management techniques. The online Appendix [42] provides more details.

6.1 Context changes

Context changes occur due to random events triggered by: 1) user mobility, modeled using mobility models such as random or Levy walk [14] and Markovian models, 2) emergency events such as detection of arrhythmia, epileptic seizure, change in mental state, and 3) user inputs, such as responding to pet's mood change. The different contexts can be represented as states in the *ContextFSM* and the events can cause state transitions (Figure 7). The events are assumed to be random with an associated probability distribution.

6.2 Experimental profiling

The available energy profiles of the scavenging sources are already obtained from [45]. We derive the power profiles of the SMDCS node for executing the BrainHealth and PetPeeves.

6.2.1 *Power Profiling*

We profiled the power consumption of several sensing systems including low capability processors such as msp430 and high end Atom processor. The power measurement setup provides the board power consumption, which includes the CPU power as well as power for driving the chipset and other associated components. We first measured the idle power of the board for each throttling mode by allowing the CPU to run idle for three minutes. Then PetPeeves and BrainHealth are executed to measure the average platform power.

The difference between the two power values gives the power consumed by the processor during the execution of the workload, which is shown in Table 7 in the online Appendix [42] for different throttling modes. The power consumption of the msp430 based motes such as TelosB and Shimmer2r were experimentally obtained by running the BSNBench benchmarking suite [46]. The benchmarking suite has specific tasks for obtaining power consumption due to computation, sensing, and communication. The sensing and computation power consumption is listed in Table 3 for benchmark signal processing applications such as Fourier transform (FFT), and peak detection. The power consumption of the Chipcon radio was measured during transmitting packets at a bit rate of 250 kbps, standard for a sensor node (www.xbow.com). The current consumption of the CC2420

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

TABLE 2
SMDCS configurations for different contexts in Ayushman (Explained in more detail in the online Appendix [42]).

Context	Requirements	Hardware Config.	Software Config.	Energy Source	Radio Protocol	Human Body
Home	Thermal safety, low energy consumption, and detection of arrhythmia within 5 seconds of onset	Shimmer ECG, and Emotive EEG, TelosB for temperature and humidity, Intel Atom	BrainHealth, secure key agreement using physiological signals (PKA)	Batteries	Bluetooth and ZigBEE, with model based communication.	Thermal dynamics
Roaming	Reliable data communication and battery less operation for 6 hrs	Shimmer ECG and accelerometer	Radio sleep scheduling, and PetPeeves	Body heat, respiration, ambulation, and sun	Retransmission and dynamic power control	Models of available energy.
Hospital	High fidelity data, thermal and drug overdose safety	Medical grade infusion pumps, pulse oximeters	Infusion control algorithm	Batteries or mains	Bluetooth, WiFi, ZigBEE, wired	Drug diffusion dynamics

TABLE 3
Sensor Power (TelosB, iMote, BSN v3, Shimmer).

Tasks	Consumed Power (mW)	Execution Time (ms)
Mean, stdev	5, 162, 6.7, 6.73	230, 220, 207, 200
FFT	5.1, 162, 6.5, 6.66	435, 102, 425, 415
Peak Detection	5.6, 156.6, 6.8, 6.6	100, 160, 90, 88

radio used in the SMDCS was measured to be 18.41 mA during transmission and 19.20 mA during reception.

6.3 Models used in Ayushman SMDCS

Power model of SMDCS: We assume that during the period of sensing $t_s = 5$ secs, the micro controller is in idle state, where it consumes P_{idle} amount of power (≈ 1 mW in TelosB motes). For a SMDCS with n nodes the sensing process can be performed in parallel by all sensors. After each sensing period the sensed data is transferred to the mobile application. During this transmission period t_{Tx} the processor is in idle state, consuming P_{idle} amount of power (approximately for 0.39 secs to transmit five seconds of 32 bit data values 60 Hz sampling rate and a transfer rate of 24 Kbps [47]). The radio transmitter will also be active during this period ($P_{radio} \approx 58$ mW being its power consumption). In a 24 hr period there will be x number of sense and transmit periods (sleep cycles) for each sensor in the SMDCS, with a duration of $(t_s + t_{Tx})$ secs each. Further, in a single day of operation of Ayushman the SMDCS nodes under go pairwise PKA execution to maintain the freshness of the encryption key among two nodes. During this execution of PKA the processor should be in active state consuming P_{PKA} amount of power for the duration of execution of the PKA algorithm t_{PKA}. The value of P_{PKA} is around 10 mW and t_{PKA} is around 1 sec as obtained from actual measurements averaged over all the commercially available platforms. Further, during the transfer of the vault ($t_{Vault} = 6.75$ s [47]), the radio is active. Thus, total energy consumption is:

$$\begin{aligned}\text{Total SMDCS Energy} &= \text{Sensing Energy} + \text{Data Transmission Energy} \\ &\quad + \text{PKA computation energy} + \text{vault transfer energy} \\ \Rightarrow E_{SMDCS} &= nx(t_s)P_{idle} + nxt_{Tx}(P_{radio} + P_{idle}) + \\ &\quad t_{PKA}P_{PKA}\left(\frac{n(n-1)}{2}\right) + t_{Vault}(P_{idle} + P_{radio})\left(\frac{n(n-1)}{2}\right) \quad (2)\end{aligned}$$

x is the number of sleep cycle to be sustained in 24 hrs.

Model of Infusion Control: Infusion pumps operate in a close loop with a networked controller to keep the drug concentration in the human blood within recommended limits. The infusion pump has three modes: a) basal, where infusion rate is I_0, b) braking, where infusion rate is a fraction f of I_0, and c) correction bolus, where infusion rate is incremented by I_b. Diffusion dynamics of the drug is spatio-temporal in nature and can be modeled using PDE Equation 3 [48].

$$\frac{\partial d}{\partial t} = \nabla(D \nabla d) + \Gamma(d_B(t) - d) - \lambda d, \quad (3)$$

where $d(x,t)$ is the tissue drug concentration at time t and distance x from the infusion site, D is the diffusion coefficient of the blood, Γ is the blood to tissue drug transfer coefficient, and $d_B(t)$ is the prescribed infusion rate at time t, and λ is the drug decay coefficient. A control algorithm in the infusion pump samples Equation 3 and adjusts the infusion levels so as to achieve the desired physiological effects while avoiding hazards such as hyperglycemia.

Model of Communication Channel: In the model based analysis phase, the Ricean flat fading model for bit error rate (BER) as a function of path loss was assumed [49], as recommended by IEEE task group 6. Eight levels of transmission power were considered for the sensor ranging from -25 dBm to 0 dBm, typical of the CC2420 radio, and the path loss was varied from 10 dB to 70 dB. Given the BER, the PDR was calculated using the equation $PDR = (1 - BER)^L$, where L is the packet length.

7 SMDCS Analysis Examples

To illustrate the analysis methodology we consider the PetPeeves application that uses a ECG sensor to continuously monitoring a user in his daily routine.

Example 1: **ECG sensor lifetime analysis:** Since long term monitoring is intended, the ECG sensor uses a model based compression technique called GeM-REM [50]. GeM-REM represents an ECG sensor using a mathematical model consisting of three Gaussian terms and stores it in both the sensor and smart phone. If the measured ECG signal matches the model then the sensor does not transmit any signal back to the base station, a smart phone. The smart phone then uses the prelearned model to regenerate the ECG signal. If the measured signal does not match the model then the sensor transmits the entire data back to the smart phone. When epilepsy occurs there is a distinct change in the ECG signal [40]. Hence, the ECG signal during an epilepsy occurrence will not match

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

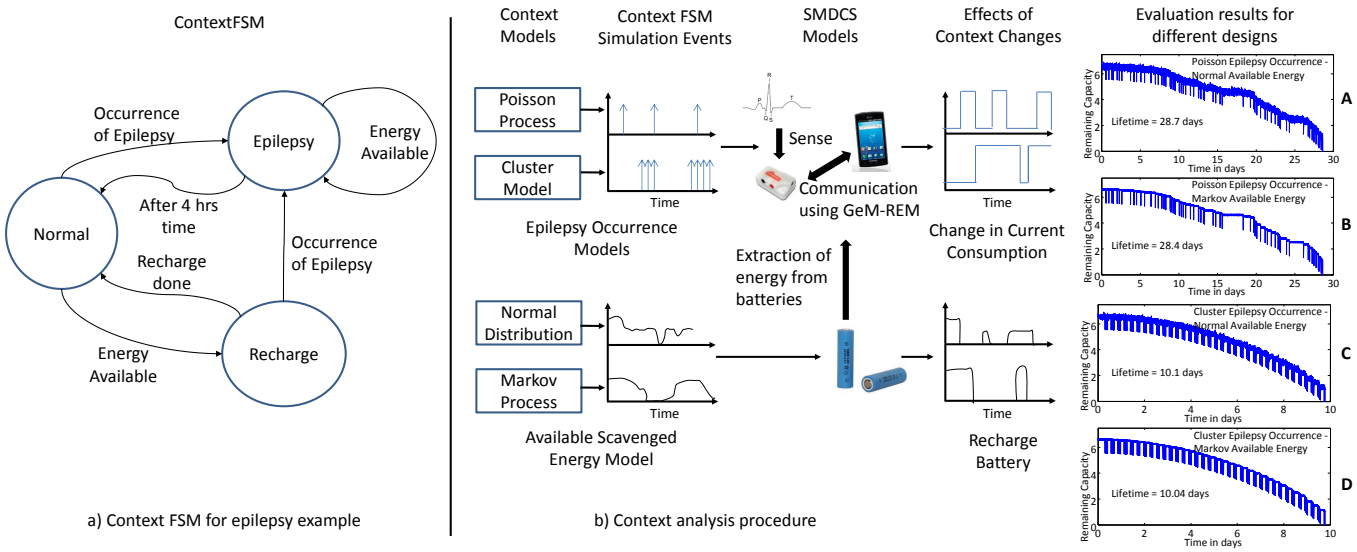

Fig. 9. **ContextFSM and the context analysis procedure for the epilepsy monitoring case study. The inputs are the ContextFSM, models of context changes, and SMDCS configurations, the outputs are the lifetime graphs for the four different model combinations. The user may choose the worst case scenario for a realistic SMDCS design.**

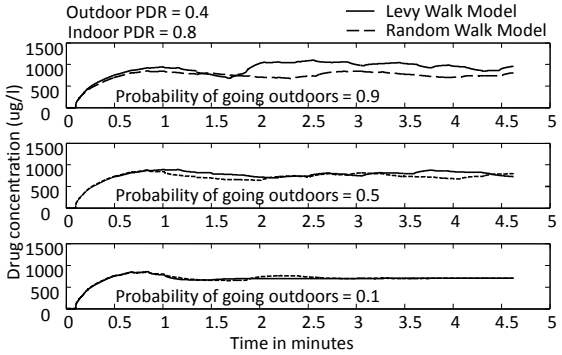

Fig. 10. **Usage of Levy walk model reveals safety flaws in infusion pump controller design.**

the pre-learned model used by GeM-REM. Thus, whenever epilepsy is detected by a detection algorithm running in the sensor side by side GeM-REM, the sensor switches mode to transmit raw ECG signals at high sampling rate of 256 Hz. Further, ECG data during epilepsy is of utmost importance for critical diagnosis. Hence, for every detection of epileptic seizure, 4 hrs of ECG data is transmitted at high sampling rate to the smart phone. At all other times GeM-REM is used for the monitoring to save transmission energy of the sensor. The sensor is operated using Lithium Ion rechargeable batteries, which are charged using energy scavenged from ambulation.

The aim of the analysis is to determine the lifetime of such a sensor and accordingly provide battery capacity to sustain the sensors for a month.

Contexts: Here there are three different context: a) ECG monitoring with GeM-REM, b) epilepsy detected and consequent high frequency signal updates, and c) energy available from scavenging sources and battery recharge. These contexts can be represented using a *ContextFSM* as shown in Figure 9a. The context changes are governed by occurrence of events. There are two events that can take place - a) occurrence of epilepsy, and b) availability of energy for recharging the battery. Epilepsy occurrence is typically modeled as a Poisson process with average occurrence of epilepsy being 0.12 per day [51]. However, recent studies conclude that epilepsy occurs in clusters, which means after the first occurrence of epilepsy subsequent occurrences are periodic [52]. The energy availability from ambulation has several models proposed by researchers. In this analysis, we consider two types of models - a) Normal, and b) Poisson distribution.

Analysis procedure: The analysis procedure is illustrated in Figure 9b. The Shimmer sensor is considered for sensing ECG. The Shimmer radio consumes 20 mA of current during transmission. The battery used in Shimmer sensors is a Lithium Ion rechargeable battery with 6500 mAh capacity and a Peukert's constant of 1.35 (http://www.shimmersensing.com/). The processes causing context changes are simulated for a time period of 30 days. Simulation of the random process describing the epilepsy occurrence outputs the number and time of epilepsy occurrences in a given day. Both the Poisson and the periodic model are simulated for this purpose. The model of the energy availability outputs the amount of energy available at a given time. The Markov process and normal distribution model were simulated. Thus this gives a total of four combinations of context models.

Whenever the *ContextFSM* is in the Normal state the Shimmer draws 1.5 mA current, which is the current consumed by the processor for executing GeM-REM and epilepsy detection code. On occurrence of epilepsy, the *ContextFSM* changes to the Epilepsy state and remains there for 4 hrs. In this state the sensor draws 20 mA current and the battery is depleted at a higher rate. Whenever, energy from the scavenging source is available, the *ContextFSM* goes to the Recharge state where the battery is charges by the amount received from the scavenging source. The simulation results at the end of the 30 day period is shown in Figure 9b. It shows that for different

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

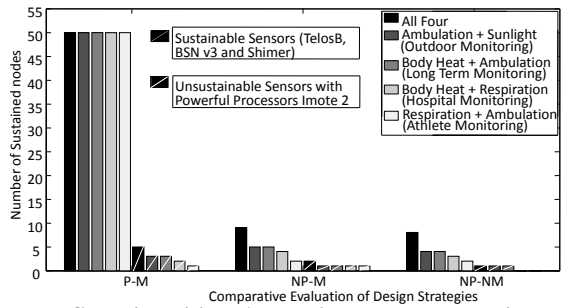

Fig. 11. **Sustainability Analysis Results - radio and processor level power management for sensors can result in better utilization of scavenging sources.**

models of context we get different lifetime of the sensor. This example thus illustrates how the context representation and analysis methodology can be used to determine the lifetime of ECG sensors in unsupervised dynamic environment.

Example 2: **Effect of context change on medical control systems** We consider the infusion pump example discussed in Section 1 and show the usage of our analysis framework. The SMDCS is in a hospital context. However, the patient wants to move around in the hospital and goes to the balcony to enjoy the view outside. This will trigger a context change in the SMDCS and the system state will transit from hospital to outdoor. In such a scenario, specifically the wireless channel properties will change resulting in a different packet delivery ratio (PDR) for the radio communication. Since the infusion pump is controlled through the wireless channel by the controller, change in the PDR may cause a drop in communication quality between the controller and the pump. Low PDR may lead to packet loss from the controller to the infusion pump. This may cause delay or loss of control inputs to the pump. In the analysis framework, two different mobility models, random and Levy walk [14], were used to simulate the context change. The hospital region was divided into two parts: indoor (PDR = 0.8) and outdoor (PDR = 0.4). The contexts were simulated for 10 cases with probability of outdoor visits varying from 0.1 to 0.9. For each sequence of control inputs the control algorithm and the pharmacokinetic model were simulated in coordination. The results of the simulation is shown in Figure 10. The results show that a random mobility pattern is less harmful (causes lower overshoots in the average case) than a Levy walk pattern. This is because in the Levy walk pattern the patient is more inclined towards an outdoor visit. However, in random walk pattern the outdoor visits are more uniformly distributed. Such complex simulation of dynamic context changes and its effect on the medical device and human body coordination cannot be performed in contemporary simulation tools and is only facilitated by our methodology.

Example 3: **Intermittent energy availability:** In this example, we consider two contexts: Home and Outdoor. The user is wearing a Shimmer mote with ECG and accelerometer sensors, and Emotiv EEG sensors to monitor his physiological and mental health through BrainHealth app and exercise performance through PetPeeves app. Energy scavenging units harvest energy from body heat, respiration, sunlight, and ambulation. When at home energy can only be scavenged from respiration. However, in outdoor environments, energy can be scavenged from all four. Table 6 in the Appendix [42] gives the available power from the scavenging sources.

TABLE 4
PDR validation results (txP - transmit power control)

Region	PDR at -25 dBm	PDR for txP
operational ICU	0.7	0.82
non-operational ICU	0.9	0.91
lobby	0.9	0.91
parking lot	0.5	0.86

A sustainability analysis plug-in was developed that used the power model of the context sensor and matched with the scavenging sources to compute the number of days a sensor can be sustained. We considered three combination of power management strategies: 1) no power management (NP-NM), 2) no processor level power management but with radio sleep scheduling (NP-M), and 3) with processor level power management and radio sleep scheduling (P-M). Figure 11 shows the time for which a SMDCS node can be sustained using the different scavenging combinations design strategies. The analysis classified the sensors in Ayushman BSN into two classes - a) sustainable sensors, such as TelosB, BSN node v3, and Shimmer, and b) unsustainable sensors, such as Imote 2, which have powerful processors (Intel XScale). In the subsequent experiments, we simulated random way point mobility model of the user and context change between home and outdoor. It was observed that under context changes the time for which the nodes can be sustained decreases to 12.27 hrs on an average, due to intermittent nature of energy availability. Further, we observed that our analysis methodology could simulate the decrease in time before recharge with reduction in the outdoor excursion frequency.

Further, if we employ the model based data communication it increases the sustainability of the sensors by a factor of 42 in case of ECG [50] and 300 in case of PPG [53]. However, a higher packet loss probability causes loss in accuracy of the model based technique. Retransmissions and transmit power control then reduce the sustainability by a factor of two.

8 VALIDATION

We consider two case studies to validate our modeling and analysis approach: 1) radio duty cycling of sensors, and 2) mobility aware transmit power control.

Validation setup: Our validation strategy consists of the following steps. We select a case study and specify the model in AADL. We then use our proposed context analysis technique to iteratively change system parameters and obtain a design that satisfies requirements. The AADL models of the SMDCSes are converted to implementations in commercially available sensors and Android smartphones using a automatic code generator, Health-Dev [38]. We empirically obtain the values of the system parameters by conducting experiments in real hospital environment. We have partnered with Phoenix St. Luke's hospital to gain access to an ICU environment.

Mobility aware transmission control: The infusion pump model used in the analysis of Example 2 is already experimentally validated [6]. Thus, we only validate whether using

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

the dynamic transmit power control algorithm discussed in Section 1 can keep the PDR at acceptable levels. In a SMDCS, PDR varies considerably due to mobility. As a strategy for increasing the PDR, a sensor can estimate the link quality in terms of path loss due to fading on each transmission. If the path loss is above a threshold then it increases the transmission power of the radio else it keeps the transmission power at the lowest value. The SMDCS system with this radio power control schedule was specified in AADL. To simulate human mobility a Levy walk model [14] was used. It was found from the simulation that the lowest transmission power level -25 dBm, gives a worst case PDR of 0.18, while a transmit power of -15 dBm gives a worst case PDR of 0.95. For the Levy walk model, with outdoor excursion probabilities ranging from 0.2 to 0.9, alternating between the -25 dBm and -15 dBm transmission levels was enough to keep a PDR at 0.88. We assume that for reliable network operation a PDR of at least 0.8 is needed. The resulting model was then implemented using Health-Dev auto code generator. We conducted experiments with the implemented model in St Luke's hospital in Phoenix Arizona. Initially we kept the transmission power to -25 dBm. We moved around on three floors which included an operational ICU cabin, non-operational ICU cabin, the lobby, and the outdoor parking lot. Table 4 gives the PDR values obtained in the different regions. In these experiments, the sensor was worn on the left pocket of the shirt while the smart phone was on the right pocket of the pant so that there is no direct line of sight communication link. Communication can only happen through mutli-path reflections (worst case scenario). The outdoor PDR in the parking lot is the lowest since there is little multi-path reflection from objects. The PDR in the operational ICU is lower since there are lot of wireless devices operating simultaneously causing interference. The lobby has the best wireless channel. When transmit power control was employed the PDR remained above 0.8.

Radio duty cycling: In Example 3, we consider that the sensor is sensing ECG signals and is performing peak detection, and FFT, representative of the signal processing involved in electrocardiogram signals [46]. The user wearing the sensors is moving from indoors and outdoors and we consider the Levy walk model of mobility for the user. The probability of the user staying indoors was set to 0.8. In the indoor state sensors can scavenge energy from respiration (1.5 W for 6 hrs) while the outdoor state sensors can scavenge from movement (1.5 W for 2 hrs) and sunlight (0.1 W for 3 hrs). We specified the power profile of the sensors and the Levy walk mobility model of the user in AADL. The power consumption of the sensor platform in radio on ($P_{ROn} + P_{proc}$) and radio off (P_{proc} only computation power) stages were obtained by performing a series of experiments on TelosB, Mica2, and Shimmer sensors. The total energy consumption of the sensor platform (E_{sensor}) in time t at a given duty cycle of $x\%$ can be obtained from Equation 2. Considering that the application has to be sustained 24 hrs from 6 hrs of scavenged energy, we varied the duty cycle in the AADL model and ran the context analyzer. We found that a duty cycle of 8.2% can be sustained using the scavenged power from respiration, walking and sunlight. The AADL model with the same radio duty cycle was provided as input to Health-Dev and the generated code was downloaded in TelosB motes. The average power consumption, measured over a single operation cycle, was 10.84 mW, which is much less than the average power available from scavenging sources ($\approx 24\ mW$). Hence, the requirements guaranteed in the analysis phase is met by the actual implementation.

9 Conclusions

In this paper, we have demonstrated a tractable randomized methodology for analyzing the effects of dynamic context changes on the interaction of SMDCS computing infrastructure with their environment. The randomized analysis can evaluate the safety and sustainability of smart mobile apps under highly probable context change sequences in polynomial time. An important extension of this work is to consider security analysis of SMDCS. We have performed initial studies on SMDCS security [54], and will consider developing a comprehensive safety, security, and sustainability analysis tool.

Acknowledgments

The authors thank Priyanka Bagade and Joseph Milazzo for the data collection. The authors also thank the editor in chief and the reviewers for their valuable comments.

References

[1] A. Banerjee and S. K. S. Gupta, "Your mobility can be injurious to your health: Analyzing pervasive health monitoring systems under dynamic context changes," in *PerCom*, March 2012, pp. 39–47.

[2] M. Joseph, P. Bagade, A. Banerjee, and S. K. S. Gupta, "bHealthy: A physiological feedback-based mobile wellness application suite," in *International conference on Wireless Health*. ACM, Nov 2013.

[3] D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky, "Formal methods based development of a pca infusion pump reference model: Generic infusion pump (gip) project," in *HCMDSS-MDPNP*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 23–33.

[4] Food and Drugs Administration, "FDA uses grammatech to analyze recalled medical devices." [Online]. Available: http://www.grammatech.com/products/codesonar/GrammaTech_FDA_Profile.pdf

[5] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2001, pp. 151–165.

[6] D. Wada and D. Ward, "The hybrid model: a new pharmacokinetic model for computer-controlled infusion pumps," *Biomedical Engineering, IEEE Transactions on*, vol. 41, no. 2, pp. 134 –142, Feb. 1994.

[7] A. Natarajan, B. de Silva, K.-K. Yap, and M. Motani, "To hop or not to hop: Network architecture for body sensor networks," in *IEEE Sensor, Mesh and Ad Hoc Communications and Networks,*, June 2009, pp. 1–9.

[8] P. W. Tuinenga, *Spice: A Guide to Circuit Simulation and Analysis Using PSpice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1991.

[9] G.-M. Elena and M. José, "ArgoSPE: Model-based software performance engineering," in *ICATPN*, 2006, pp. 401–410.

[10] P. Vibha, T. Yan, P. Jayachandran, Z. Li, S. H. Son, J. A. Stankovic, J. Hansson, and T. Abdelzaher, "ANDES: An analysis-based design tool for wireless sensor networks," in *Real-Time Systems Symposium, RTSS 2007. 28th IEEE International*, pp. 203–213.

[11] Z. Jiang, M. Pajic, and R. Mangharam, "Model-based closed-loop testing of implantable pacemakers," in *Proceedings of the IEEE/ACM Second International Conference on Cyber-Physical Systems*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 131–140.

[12] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, "Toward patient safety in closed-loop medical device systems," in *ACM/IEEE International Conference on Cyber-Physical Systems*. New York, NY, USA: ACM, 2010, pp. 139–148.

[13] Codeproanalytics. https://marketplace.eclipse.org/content/codepro-analytix.

[14] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the levy-walk nature of human mobility," in *INFOCOM*, April 2008, pp. 924 –932.

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2014.2334606, IEEE Transactions on Mobile Computing

[15] J. C. Jensen, D. Chang, and E. A. Lee, "A model-based design methodology for cyber-physical systems," in *Wireless Communications and Mobile Computing Conference,*, July 2011, pp. 1666 – 1671.

[16] G. Frehse, "Phaver: Algorithmic verification of hybrid systems past hytech," in *HSCC*, 2005, pp. 258–273.

[17] Modelica and the Modelica Assoc, "Modelica," https://modelica.org/.

[18] P. Joshi, C.-S. Park, K. Sen, and M. Naik, "A randomized dynamic program analysis technique for detecting real deadlocks," in *ACM SIGPLAN PLDI*, ser. PLDI '09. New York, NY, USA: ACM, 2009, pp. 110–120.

[19] J. Hu, J. Lygeros, and S. Sastry, "Towards a theory of stochastic hybrid systems," *Hybrid Systems: Computation and Control*, vol. 1790, pp. 160–173, 2000.

[20] Altera, "VHDL," http://www.altera.com/support/examples/vhdl/vhdl.html.

[21] SAE, "Advanced architecture design language," http://www.aadl.info/aadl/currentsite/.

[22] OMG, "Unified modeling language," http://www.uml.org/.

[23] C.-L. Fok, A. Petz, D. Stovall, N. Paine, C. Julien, and S. Vishwanath, "Pharos: A testbed for mobile cyber-physical systems," *Univ. of Texas at Austin, Tech. Rep. TR-ARiSE-*, 2011.

[24] D. Acharya, V. Kumar, and H.-J. Han, "Performance evaluation of data intensive mobile healthcare test-bed in a 4g environment," in *2nd ACM international workshop on Pervasive Wireless Healthcare*. New York, NY, USA: ACM, 2012, pp. 21–26.

[25] Mathworks, "MATLAB ® and Simulink," http://www.mathworks.com/.

[26] SysML, "Systems modeling language (sysml)," http://www.sysml.org/.

[27] "Flovent." [Online]. Available: http://www.mentor.com/products/mechanical/products/flovent

[28] A. Cervin and K.-E. Arzen, *Model-Based Design for Embedded Systems*. CRC Press, 2011, ch. TrueTime: Simulation Tool for Performance Analysis of Real-Time Embedded Systems, pp. 93–119.

[29] K. A. Aalborg, K. E. Andersen, and M. Hjbjerre, "A Bayesian Approach to Bergman's Minimal Model," in *in: C.M. Bishop, B.J. Frey (Eds.), Proceedings of the 9th Intl. Workshop on Artificial Intelligence,*, 2003.

[30] A. Banerjee, S. Kandula, T. Mukherjee, and S. K. S. Gupta, "Band-aide: A tool for cyber-physical oriented analysis and design of body area networks and devices," *ACM Trans. Embed. Comput. Syst.*, vol. 11, no. S2, pp. 49:1–49:29, Aug 2012.

[31] W. Yan, Y. Xue, X. Li, J. Weng, T. Busch, and J. Sztipanovits, "Integrated simulation and emulation platform for cyber-physical system security experimentation," in *Proceedings of the 1st international conference on High Confidence Networked Systems*, ser. HiCoNS '12. New York, NY, USA: ACM, 2012, pp. 81–88.

[32] A. Bhave, B. Krogh, D. Garlan, and B. Schmerl, "Multi-domain modeling of cyber-physical systems using architectural views," in *Proceedings of the 1st Analytic Virtual Integration of Cyber-Physical Systems Workshop.*, 30 November 2010.

[33] D. Henriksson and H. Elmqvist, "Cyber-physical systems modeling and simulation with Modelica," in *Proceedings of the 8th International Modelica Conference, Dresden, Germany*, 2011, pp. 502–509.

[34] K. Bauer, "A new modelling language for cyber-physical systems," Ph.D. dissertation, Department of Computer Science, University of Kaiserslautern, Germany, Kaiserslautern, Germany, January 2012.

[35] W. Reisig, *Petri nets: an introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.

[36] A. Banerjee and S. K. S. Gupta, "Spatio-temporal hybrid automata for safe cyber-physical systems: A medical case study," *Intl' Conf' on Cyber-Physical Systems*, pp. 71–80, 2013.

[37] A. Platzer, "Quantified differential dynamic logic for distributed hybrid systems," in *CSL*, ser. LNCS, A. Dawar and H. Veith, Eds., vol. 6247. Springer, 2010, pp. 469–483.

[38] A. Banerjee, S. Verma, P. Bagade, and S. K. S. Gupta, "Health-dev: Model based development pervasive health monitoring systems," *BSN,*, pp. 85–90, 2012.

[39] F. Adelstein, S. K. S. Gupta, G. G. Richard, and L. Schwiebert, *Fundamentals of Mobile and Pervasive Computing*. McGraw-Hill, Dec 2004.

[40] H. Veld and M. Ordelman, "Context aware algorithm for epileptic seizure detection," in *Awareness deliverables*, 2005.

[41] H. U. Gerber and S.-Y. R. Li, "The occurrence of sequence patterns in repeated experiments and hitting times in a markov chain," *Stochastic Processes and their Applications*, vol. 11, no. 1, pp. 101 – 108, 1981.

[42] A. Banerjee and S. K. S. Gupta, "Analysis of smart mobile applications for healthcare under dynamic context changes," http://impact.asu.edu/publication/TMCAppendix.pdf.

[43] K. K. Venkatasubramanian, G. Deng, T. Mukherjee, J. Quintero, V. Annamalai, and S. K. S. Gupta, "Ayushman: A Wireless Sensor Network Based Health Monitoring Infrastructure and Testbed," in *Distributed Computing in Sensor Systems*, July 2005, pp. 406–407.

[44] S. K. S. Gupta, T. Mukherjee, and K. K. Venkatasubramanian, *Body Area Networks: Safety, Security, and Sustainability*. Cambridge University Press, 2013.

[45] J. A. Paradiso, "Energy scavenging for mobile and wireless electronics," *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 18–27, Jan.-March 2005.

[46] S. Nabar, A. Banerjee, S. K. S. Gupta, and R. Poovendran, "Evaluation of body sensor network platforms: a design space and benchmarking analysis," in *Wireless Health*. New York, NY, USA: ACM, 2010, pp. 118–127.

[47] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Green and sustainable cyber-physical security solutions for body area networks," in *BSN*. IEEE Computer Society, 2009, pp. 240–245.

[48] T. L. Jackson and H. M. Byrne, "A mathematical model to study the effects of drug resistance and vasculature on the response of solid tumors to chemotherapy," *Mathematical Biosciences*, vol. 164, no. 1, pp. 17 – 38, 2000.

[49] "http://math.nist.gov/mcsd/savg/papers/15-08-0780-09-0006-tg6-channel-model.pdf."

[50] S. Nabar, A. Banerjee, S. K. S. Gupta, and R. Poovendran, "GeM-REM: Generative model-driven resource efficient ecg monitoring in body sensor networks," in *Body Sensor Networks (BSN),*, May 2011, pp. 1 –6.

[51] J. G. Milton, J. Gotman, G. M. Remillard, and F. Andermann, "Timing of seizure recurrence in adult epileptic patients: A statistical analysis," *International League Against Epilepsy*, 1987.

[52] S. R. Haut, "Seizure clustering," *Epilepsy Behavior*, vol. 8, no. 1, pp. 50 – 55, 2006.

[53] S. Nabar, A. Banerjee, S. K. S. Gupta, and R. Poovendran, "Resource-efficient and reliable long term wireless monitoring of the photoplethysmographic signal," in *Wireless Health*, October 2011, pp. 1 –6.

[54] P. Bagade, A. Banerjee, J. Milazzo, and S. K. Gupta, "Protect your bsn: No handshakes, just namaste!" in *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, May 2013, pp. 1–6.

Ayan Banerjee , Ph.D., is postdoctoral fellow at the IMPACT Lab. His research interest include cyber-physical system design and analysis in several domains such as healthcare and cloud computing. More information on his research is available at http://impact.asu.edu/ayan/index.html.

Sandeep K.S. Gupta , Ph.D., is the director of IMPACT Lab and is also the chair of computer science and engineering at ASU. His current research focuses on dependable, criticality-aware, adaptive distributed systems with emphasis on wireless sensor networks, thermal and power-aware computing and communication, and pervasive healthcare. He has co-authored the book Fundamentals of Mobile and Pervasive Computing, McGraw Hill.

1536-1233 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.