Healthopia: Towards Your Well-Being in Everyday Life

(Invited Paper)

Chulhong Min, Chungkuk Yoo, Youngki Lee, Junehwa Song
Computer Science, KAIST
Daejeon, 305-701, Republic of Korea
{chulhong, ckyoo, youngki, junesong}@nclab.kaist.ac.kr

ABSTRACT

In this paper, we propose Healthopia, a full-fledged health monitoring platform. With the prevalence of smartphones and wearable health sensors, *sensor-enabled mobile well-being applications* emerge, for instance, calorie monitor [13], fall detector [11], pollution detector, heartbeat monitor [1][4]. Despite the importance and usefulness of such applications, it is significantly challenging to develop such applications; developers should concern about many complicated issues, such as accurate data analysis and inferring, distributed programming over multiple sensing devices, resource optimization. To address such issues, and facilitate the development and deployment of the applications, we develop Healthopia platform. Using the declarative query-based API, applications can easily delegate monitoring requests for diverse health information, e.g., calorie consumption, heart condition, pollution level. Healthopia takes care of complicated underlying issues in sensor data processing and resource management, and delivers the requested health information. In this paper, we propose a novel sensor control technique applied in Healthopia, which reduces battery power significantly.

Categories and Subject Descriptors

K.8 [**Personal Computing**]: General; C.5.3 [**Microcomputers**]: Portable devices; C.3.3 [**Special-Purpose and Application-based System**]: Microprocessor/microcomputer applications

General Terms

Design, Experimentation, Management, Measurement, Performance.

Keywords

Mobile computing, Healthcare application, Well-being, Context monitoring, Resource saving, Smartphone, Wearable sensors.

1. INTRODUCTION

In metropolitan cities, people increasingly pursue *well-being* and take care of their health in their everyday lives. They are willing to deeply understand their body status, e.g., from detailed calorie consumption records to subtle abnormality in their heart conditions. In addition, people start to pay more attention to diverse health-affecting factors of surrounding spaces, for instance, dust level, CO_2 level and humidity. The advent of pervasive computing technologies opens up a new opportunity to satisfy the people's ever increasing desire and needs for everyday well-being and healthcare. *Sensor-enabled mobile well-being applications* will be a key means; they provide *personalized* services in-situ by monitoring individual's health status and surroundings through always-carrying smartphones. For example, a caloric expenditure monitor notifies people of calorie consumption involved in physical activities and encourages them to take more exercise [3][13]. A heartbeat monitor [1][4] is able to detect irregular heart behavior in advance using wearable biomedical sensors such as electrocardiogram (ECG) and blood volume pulse (BVP) sensors.

Developing such applications, however, often involves multi-lateral challenges and requires huge effort from developers. First, inferring health information from raw sensor data requires highly complex processing logics and learning processes. For example, caloric measurement is inferred by accurate learning-based recognition of user's physical activities from raw acceleration sensing data [13]. Second, applications often require carefully coordinated, distributed design over smartphones and heterogeneous wearable sensing devices. Good application design is significantly difficult, since programmers should have deep understandings on multiple operating systems and programming languages, for instance Java and Android for smartphones and NesC and TinyOS for sensing devices. Finally, even more challenging, developers should highly optimize resource use of applications. Smartphones and wearable health sensors have scarce battery powers and limited computation capacity, which are lacking to support high-rate sensing and complicated processing for health monitoring. Without proper optimization, applications may shut down quickly due to fast battery drain. Also, only few applications might run due to conflicts in CPU consumption.

In this paper, we envision Healthopia, a full-fledged health monitoring platform. It facilitates development and deployment of diverse mobile well-being applications over smartphones and wearable health sensors. With the system support, individual applications need to simply delegate health monitoring requests to Healthopia, and no longer have to concern about the complex issues involved in health monitoring. Instead, Healthopia takes charge of underlying issues for health monitoring; from accurate data analysis and inferring, to resource optimization, and to private data management and delivery. We implement and deploy the Healthopia prototype over the off-the-shelf smartphones (Nexus Ones) and diverse wearable sensor devices with different sensing modules such as BVP, ECG, Galvanic Skin Response (GSR), dust, and CO_2 sensors and accelerometers. We are continuously addressing important challenges and extending our

ISABEL '11, October 26-29, Barcelona, Spain

platform to build a highly practical and useful health monitoring platform.

Among many important issues, in this paper, we focus on a key challenge, the resource scarcity problem, especially shortage of battery power and inter-sensor bandwidth. Wearable sensors are preferred to use a small size battery for better wearability, and consequently have very limited battery capacity. For example, a 2.5cm X 1.2cm Li-ion battery, one of the most widely used for the sensors, has only 250mA capacity. Our experiment shows that it only supports several hours of dust level sensing. For wireless communication, ZigBee is widely used as a communication channel of wearable sensors for energy efficiency and large coverage, but guarantees reliable communication under only about 50 packets per second. However, health monitoring often requires high-rate transmission. For example, ECG sensor and accelerometer need to transmit sensing data at 10-100Hz. Even more serious, a number of healthcare applications often need to share highly scarce resources of the smartphone and the sensors.

To address the challenge, we develop a novel sensor control technique. We observe that many mobile well-being applications specify their monitoring queries in the conjunctive form of multiple contexts. For example, a heart monitoring application can register a query, 'let me know if a user's heart rate exceeds over some threshold when she is jogging outside in a sunny day'. The key idea of the sensor control technique is to turn off a partial set of sensor devices involved in the context monitoring, utilizing the unique characteristics of the query structure. For instance, in the above example, if the system knows that she is not moving, it does not need to monitor other contexts such as user's heart rate, location, and temperature, and thus, can turn off the associated sensor devices. Through the deactivation of non-essential sensors, the technique significantly reduces battery consumption of sensors and bandwidth usage between the smartphone and sensors.

The contributions of this paper are as follows. First, we envision Healthopia, a health monitoring system to support mobile well-being applications. Second, we show architecture overview of the Healthopia prototype system and Healthopia APIs for application development. Third, we present a novel sensor control technique over a smartphone and wearable health sensors. It significantly reduces energy consumption and bandwidth usage without decreasing of quality of applications. Finally, we show resource benefits of our sensor control technique through experiments.

The rest of this paper is organized as follows. In Section 2, we present related work. Section 3 describes the architecture overview of Healthopia and Healthopia APIs. We then explain the sensor control technique in Section 4, and shows experimental results in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORK

Various healthcare applications have been proposed for patients in hospitals or having chronic diseases, and the elderly. Example applications include fall detector [11], heartbeat monitor [1][4], diabetes manager [16], children behavior monitor [6] and eye tracker for medical applications [18]. Most prior works focus on accurate retrieval of target health information. They do not consider system issues involved in the application deployment for daily use, which is critical for mobile well-being applications. In addition, while previous works monitor application-specific health information fitting to its purposes, Healthopia supports diverse healthcare applications as a common platform by providing high-level health monitoring APIs and resource-efficient.

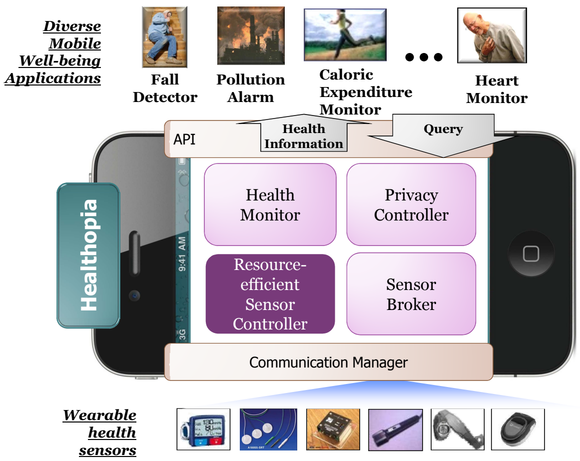


Figure 1. Architecture of Healthopia prototype

Some techniques have been proposed to utilize and incorporate wearable sensors for health monitoring applications, for instance time synchronization between multiple sensors [17] and wireless protocol design [2]. Unlike Healthopia, they do not deal with a smartphone environment. Also, energy issue is not discussed, which is the most important one for daily use of mobile well-being applications.

Energy optimization and management are one of the most critical issues for battery-powered mobile and sensor devices, due to the scarcity of their battery capacity. Several sensor systems such as Pixie [15], Eon [19], and Level [12] have been proposed to efficiently manage energy use of a single sensor device. Also, systems such as DSOM [5] have been proposed to manage the energy use of mobile devices, e.g., smartphones. Such diverse energy management and reduction techniques can be complimentarily used in Healthopia to further reduce energy use. The proposed sensor control technique significantly reduces energy use over wearable health sensors, utilizing a new, unique characteristic of health monitoring queries.

Our prior works, SeeMon [8][9],Orchestrator [10] and MobiCon [7] have been proposed to support general context monitoring applications on a smartphone and sensors. Healthopia can be considered as a specialized platform targeting *mobile well-being applications*. We envision that Healthopia takes more specific considerations on the characteristics of the applications such as data integrity, accurate inferring, health context privacy, and application priority. In the design of system architecture, such factors have been newly considered; for instance, application priority is handled as an important aspect in the design of the sensor control technique.

3. HEALTHOPIA PROTOTYPE

3.1 Architecture Overview

In this section, we present the system architecture of Healthopia. Figure 1 shows core functionalities of Healthopia and its interactions with the mobile well-being applications and wearable health sensors. The application requests the health information of interest to the system through API. This enables application developers to be freed from much effort for resource management,

Table 1 Context type and value

Context Type	Context Value
Heartbeat	dangerous, rapid, normal, slow, too slow
Activity	walking, standing, running, sitting
Dust	danger, terrible, bad, sensitive, normal, good
CO_2	worst, pretty bad, a little bad, allowed, normal
Temperature	terribly hot, hot, warm, cool, cold
Humidity	very dry, dry, humid, very humid
Light	very light, light, dark, very dark
Location	CS building, restaurant, home, …
Time	morning, afternoon, evening, night

inference of health information, and dynamic discovery of wearable health sensors. Healthopia continuously processes sensor data and extracts the health information specified as queries, and delivers the results to the associated applications. Meanwhile, it also manages the communication with the sensors in the background. Due to the dynamics of the sensors, Healthopia continuously monitors the presence and resource availability of the sensors and adapts its operations to such dynamic changes.

Healthopia has four components which are essential to support mobile healthcare applications. The *Health Monitor* is responsible for health information monitoring. It continuously processes sensor data and recognizes the health information. Healthopia provides the applications with health contexts by adopting diverse feature extractions and classifiers such as FFT and GMM. Table 1 shows the example of context types and its values. The *Resource-efficient Sensor Controller* manages the resources of the sensors. It selects the essential sensors for monitoring the queries and resolves the conflicts of the applications for a shared resource (See Section 4). *Privacy Controller* manages a user's privacy for her health information. It enables the user to select the trusted applications and prohibits untrusted applications from accessing her health information. Various privacy research and techniques can be further incorporated into Healthopia [14]. *Sensor Broker* manages wearable health sensors. By periodically broadcasting the beacon message, it dynamically discovers available nearby sensors. It also interprets sensor data and sensor control messages.

3.2 Healthopia API

Healthopia provides application developers with APIs to monitor health contexts of interest through *health monitoring query*. The query semantics is inspired by the Context Monitoring Query which is presented in our previous works [8][9]. The key extension is the application priority for resource use. Different well-being applications often have different priority for resource use. For example, for the elderly, the fall detector has a higher priority than the calorie monitor.

```
Monitor  <context element>
               (AND <context element>)*
Priority <priority> (0, 1, …, 9, 9 is the highest)
```

The query language consists of two fields; *Monitor* and *Priority*. The *Monitor* field describes health-related contexts which the applications want to monitor. Such contexts are specified as a conjunctive normal form (CNF) of context elements; a context element is written as <*context type* == *context value*>. The *Priority* field specifies the importance of the associated query when multiple queries conflicts for a shared resource.

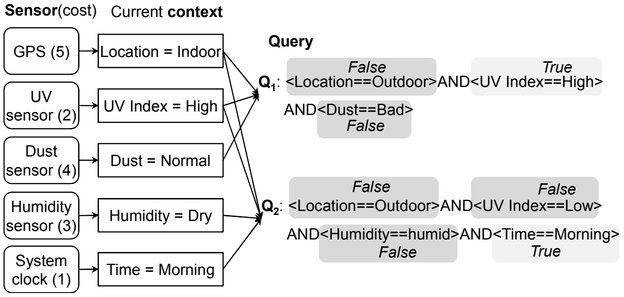


Figure 2. Example situation

For example, when a UV index monitoring application is running for a woman in her twenties, it wants to monitor if the UV index is high when she is outside in daytime. The application registers a query as follows:

```
Monitor: <location == outdoor> AND
 <time == daytime> AND <UV index == high>
Priority: 7
```

4. RESOURCE-EFFICIENT SENSOR CONTROL

To achieve resource efficiency for mobile well-being applications, we develop a novel sensor control technique. The key idea of the technique is to turn off a partial set of sensors for the context monitoring. It is enabled by the unique characteristic of the health monitoring query, a CNF form; a single false-state element, a context element evaluated as false, leads the query state to false. For instance, to determine the state of the example query described in Section 3.2, it is necessary to monitor three contexts; her location, the time of day, and the UV index. If in the evening, the query state can be determined as false without monitoring her location and the UV index. In this case, GPS and UV sensors can be turned off.

If multiple context elements of a query are evaluated as false, any one context element is needed to be monitored until it becomes true. It is important to carefully select a false-state element to monitor. Depending on the selection, different sets of associated sensors will be turned off, thereby resulting in different level of resource benefit. For instance, in the above example, if she enters a building, two context elements of the query, i.e., about time and location, become false. In this case, it is more beneficial to select the element for time condition. This is mainly because turning off GPS, a power hungry sensor is more beneficial than turning off the system clock.

The sensor control technique is a problem to find a partial set of sensors that maximizes the resource saving while supporting the registered queries. The technique solves the problem as follows; (1) it enumerates all possible candidate sensor sets, each of which can determine the state of all registered queries, (2) it selects the sensor set with the minimum resource consumption. To quantify the resource consumption of a sensor, we use a weighted cost function as follows:

$$cost(sensor) = \alpha\times energy + \beta\times bandwidth + \chi\times CPU + \ldots$$

Healthopia enables diverse resource saving policies by specifying the coefficients of the cost function. For example, for the policy that minimizes the energy consumption, the alpha is set to one while the others are set to zero.

Table 2 Energy profiles

Sensor	Context	Task		Energy Demand (mJ/s)	Baseline (mJ/s)
		Sensing	Transmission		
CO_2	CO_2 level	0.33	0.33	120.3	34.4
Dust	Dust level	0.5	0.5	500.6	
ECG	Heartbeat	10	1	58.7	
Temperature Light Pressure Acceleration	Weather	0.2	0.2	55.3	
GPS	Location	0.2	0.2	231.9	

Table 3 Simulation setup

Parameter	Default Value
Number of queries	5
Number of elements per query	3
Number of context	5
Number of sensors	5
Base energy consumption	34.4
Energy demand for monitoring	55.3 – 500.6
Number of iterations	100

We describe the detailed process of the technique with the example shown in Figure 2. Two queries, Q_1 and Q_2, are registered and use five sensors for health monitoring. In the current context, Q_1 has two false-state context elements and Q_2 has three. The technique enumerates six candidate sensor set; {GPS}, {GPS, UV}, {GPS, humidity}, {dust, GPS}, {dust, UV}, and {dust, humidity}. Among the candidates, the best sensor set is {GPS} with the minimum cost, 5.

Healthopia considers the priority of the query by adopting the cost function as follows:

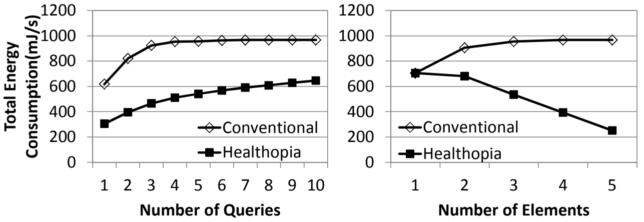
$$cost'(sensor) = cost(sensor) \times priority(sensor)$$

The priority of the sensor is determined as the maximum priority of the queries which use the sensor.

5. EVALUATION

5.1 Energy Profiling for Health Monitoring

In Healthopia, energy benefit comes from the avoidance of unnecessary task execution on sensor motes such as sensing data, processing, and data transmission. In order to estimate the benefit in terms of battery power, we measure the energy consumption of sensor motes for diverse health monitoring using a digital multimeter (Agilent 34410A) similar to in [5]. For sensor motes, we use off-the-shelf ZigbeX motes (MicaZ clone). They incorporate a daughter board for diverse sensing modalities, and are equipped with Atmega 128L MCU, CC2420 RF transceiver, and TinyOS 1.1.1 as an operation system. Table 2 shows the example profiling results. For instance, the first row shows the energy demand for CO_2 monitoring and its baseline. The baseline represents the energy consumption for primitive operations of sensor motes without the execution of health monitoring. If CO_2 sensor is not necessary to determine the state of associated queries, we can reduce the energy consumption from 120.3mJ/s to 34.4mJ/s.



(a) Effect of number of queries (b) Effect of number of elements

Figure 3. Energy benefit of Healthopia

5.2 Energy Benefit of Healthopia

In this section, we present the energy efficiency of Healthopia through simulation using energy profiles presented in Section 5.1. It is very difficult to analyze the overall evaluation in real setting, because the efficiency highly depends on various system parameters such as the number of the queries, the number of the sensors, and sensing data value. As shown in Table 3, we carefully set the simulation parameters reflecting realistic environment. We generate the queries by randomly choosing the requested context. For smoothing the result, we perform 100 iterations for each measurement. The metric of evaluation is the total sum of the energy consumption of all sensor motes (mJ/s). For each iteration, we measure the energy consumption for all possible combinations of query states and then, take the average energy consumption over the experiments. We compare Healthopia with a conventional context monitoring system which turns on all necessary sensors for currently registered queries. Note that the total energy consumption when all sensors execute monitoring task is 966.8mJ/s.

Figure 3 (a) describes the evaluation results while varying the number of queries. The energy consumption of both the conventional system and Healthopia increases as the number of queries increases. This is obvious because more queries require more wearable health sensors to determine the final states. However, Healthopia shows more energy-efficient monitoring than the conventional engine, since it avoids unnecessary sensing and transmission by adopting the resource-efficient sensor control technique. To support all queries, Healthopia consumes 646.2mJ/s whereas the conventional system does 966.8mJ/s.

Figure 3 (b) shows the energy consumption as the number of elements per query increases. The energy consumption of Healthopia decreases as the number of elements increases whereas that of the conventional system increases. Specifically, for the queries with five elements, Healthopia consumes just 251.0mJ/s while the conventional system does 966.8mJ/s. This is because more number of elements gives more opportunity to save energy consumption; monitoring one false-state context element is enough to determine the state of query. In the case that a query contains only one element, Healthopia consumes the same amount of energy as the conventional system does.

6. CONCLUSION

In this paper, we present Healthopia, a full-fledged health monitoring platform over smartphones and wearable health sensors. Healthopia takes care of many complicated service issues and enables application developer to easily develop diverse mobile well-being applications. Through the resource-efficient sensor control technique, Healthopia further achieves a high level of resource efficiency without the sacrifice of application quality. We implement a prototype of Healthopia system on an off-the-shelf smartphone, Nexus One, and wearable health sensors

(MicaZ clones). We also show the resource efficiency and feasibility of Healthopia through experiments.

7. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (2011-0018264).

8. REFERENCES

[1] Censi, F., Calcagnini, G., Mattei, E., Triventi, M., and Bartolini, P. Simulation of daily ECG monitoring strategies for atrial fibrillation patients. In *Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (Rome, November 07-10, 2010). ISABEL'10. 1-2.

[2] Chen, W., Nguyen, S. T., Coops, R., Oetomo, S. B., and Feijs, L. Wireless transmission design for health monitoring at neonatal intensive care unit. In *Proceedings of the 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (Bratislava, November 24-27, 2009). ISABEL'09. 1-6.

[3] Consolvo, S., McDonald, D. W., Toscos, T., Chen, M. Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., and Lanay, J. A. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (Florence, Italy, April 05-10, 2008). CHI '08. ACM, New York, NY, USA, 1797-1806.

[4] De Chazal, P. and Reilly, R. B. A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features. *IEEE Transactions on Biomedical Engineering*. 53, 12 (Dec. 2006), 2535-2543.

[5] Fei, Y., Zhong, L., and Jha, N. K. An energy-aware framework for dynamic software management in mobile computing systems. *ACM Trans. Embed. Comput. Syst.* 7, 3, Article 27 (May 2008).

[6] Hwang, I., Jang, H., Nachman, L., and Song, J. Exploring inter-child behavioral relativity in a shared social environment: a field study in a kindergarten. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (Copenhagen, Denmark, September 26-29, 2010). Ubicomp '10. ACM, New York, NY, USA, 271-280.

[7] Kang, S., Iyengar, S. S., Lee, Y., Min, C., Ju, Y., Park, T., Lee, J., Rhee, Y., and Song, J. MobiCon: Mobile Context Monitoring Platform for Sensor-Rich Dynamic Environments, to appear in *Communications of the ACM*.

[8] Kang, S., Lee, J., Jang, H., Lee, H., Lee, Y., Park, S., Park, T., and Song, J. SeeMon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proceeding of the 6th international conference on Mobile systems, applications, and services* (Brekenridge, Colorado, USA, June 17-20, 2008). MobiSys '08. ACM, New York, NY, USA, 267-280.

[9] Kang, S., Lee, J., Jang, H., Lee, Y., Park, S., and Song, J. A Scalable and Energy-efficient Context Monitoring Framework for Mobile Personal Sensor Networks, *IEEE Transactions on Mobile Computing*, 9, 5(May 2010), 686-702.

[10] Kang, S., Lee, Y., Min, C., Ju, Y., Park, T., Lee, J., Rhee, Y., and Song, J. Orchestrator: An active resource orchestration framework for mobile context monitoring in sensor-rich mobile environments. In *Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications* (Mannheim, Germany, March 29 – April 02, 2010). PerCom'10. 135-144.

[11] Kangas, M., Konttila, A., Winblad, I., and Jamsa, T. Determination of simple thresholds for accelerometry-based parameters for fall detection. *IEEE Proc. on Engineering in Medicine and Biology Society* (Lyon, August 22-26, 2007). EMBS'07. 1367-1370.

[12] Lachenmann, A., Marrón, P. J., Minder, D., and Rothermel, K. Meeting lifetime goals with energy levels. In *Proceedings of the 5th international conference on Embedded networked sensor systems* (Sydney, Australia, November 06-09, 2007). SenSys '07. ACM, New York, NY, USA, 131-144.

[13] Lester, J., Hartung, C., Pina, L., Libby, R., Borriello. G., and Duncan, Glen. Validated caloric expenditure estimation using a single body-worn sensor. In *Proceedings of the 11th international conference on Ubiquitous computing* (Orlando, Florida, USA, September 30 – October 03, 2009). UbiComp'09. ACM, New York, NY, USA. 225-234.

[14] Lioudakis, G. V., Lamprinakos, G. and Kosmatos, E. and Kaklamani, D.I. and Venieris, I. S. Introducing privacy-awareness in remote healthcare monitoring. In *Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (Rome, November 07-10, 2010). ISABEL'10. 1-5.

[15] Lorincz, K., Chen, B., Waterman, J., Werner-Allen, G., and Welsh, M. Resource aware programming in the Pixie OS. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (Raleigh, North Carolina, USA, November 05-07, 2008). SenSys '08. ACM, New York, NY, USA, 211-224.

[16] Nachman, L., Baxi, A., Bhattacharya, S., Darera, V., Dechpande, P., Kodalapura, N., Mageshkumar, V., Rath, S., Shahabdeen, J., and Acharya, R. Jog Falls: A Pervasive Healthcare Platform for Diabetes Management. In *Proceedings of the Eighth International Conference on Pervasive Computing* (Helsinki, Finland, May 17-20, 2010). Pervasive'10. 94-111.

[17] Nakamura, M., Nakamura, J., Shuzo, M., Warisawa, S., and Yamada, I. Collaborative processing of wearable and ambient sensor system for health monitoring application. In *Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (Rome, November 07-10, 2010). ISABEL'10. 1-5.

[18] Raudonis, V., Simutis, R., and Narvydas, G. Discrete eye tracking for medical applications. In *Proceedings of the 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (Bratislava, November 24-27, 2009). ISABEL'09. 1-6.

[19] Sorber, J., Kostadinov, A., Garber, M., Brennan, M., Corner, M. D., and Berger, E. D. Eon: a language and runtime system for perpetual systems. In *Proceedings of the 5th international conference on Embedded networked sensor systems* (Sydney, Australia, November 06-09, 2007). SenSys '07. ACM, New York, NY, USA, 161-174.