



NRC Publications Archive Archives des publications du CNRC

WSN-based real time data collection, independent living spaces, event-driven architecture, web services

Yang, Fujun; Xue, Henry; Shen, Weiming; Xue, Yunjiao; Hao, Qi

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1109/ICSMC.2011.6083995>

Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2640-2645, 2011-11-21

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=84a374a1-d01f-43e4-a1bd-b10871970788>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=84a374a1-d01f-43e4-a1bd-b10871970788>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



WSN-based Real-time Data Collection in Independent Living Spaces

Fujun Yang, Henry Xue, Weiming Shen, Yunjiao Xue, Qi Hao

Centre for Computer-assisted Construction Technologies, National Research Council, London, Ontario, Canada
[fujun.yang; henry.xue; weiming.shen; yunjiao.xue; qi.hao]@nrc.gc.ca

Abstract— *Collecting and analyzing real-time data of a distributed wireless sensor network system is very challenging since it combines a series of processes to continuously monitor and respond to a wide variety of events. This paper presents an event-driven, real-time data collection system architecture for an independent living spaces monitoring system. A service-replacement-based timestamp matching structure is also proposed to improve the system's robustness and maintainability. A prototype system has been developed to validate the proposed approach.*

Keywords—*wireless sensor networks; real-time data collection; independent living spaces; event driven architecture; web services*

I. INTRODUCTION

More and more research groups in both academia and industry have started working on wireless sensor networks (WSNs) and their applications. Some of them applied WSNs to the pervasive computing domain [1]. Modern sensors have the capability of quickly responding to rapid changes in environments and sending sensed data to processing centers in areas where cabling is inappropriate [2]. This inherent advantage of WSNs has been widely applied to various monitoring systems. However, most traditional monitoring systems are not scalable when the number of objects increases. Integrating numerous heterogeneous elements is a manual and ad hoc process for a traditional monitoring system [1]. In order to dynamically add or remove monitoring targets, to extend the computing capability, and to reduce integration cost, we proposed a service-oriented integration framework for independent living spaces (ILS) [3]. A dynamic, loosely-coupled, and scalable monitoring system for ILS is a typical real-time system. Since a real-time system usually has two types of events: hard real-time events (where responding time has a significant impact on the service quality) and soft real-time events (where responding time has minimal impact on the service quality), it is difficult to effectively collect data from an integrating system, especially under a wirelessly distributed dynamic environment [4]. How to integrate the infrastructure (device layer), middleware, and services, and how to provide users with appropriate services through the Internet, are still research challenges.

The combination of smart household appliances, wireless sensor networks, and service-oriented computing will change traditional monitoring modes and provide a kind of ubiquitous surveillance to help people (especially seniors) live independently and safely in their own homes.

Real-time system performance is affected by two important factors [4, 5]: one is to collect data in a timely manner; the other is system fault tolerance management. According to the requirements of monitoring independent living spaces [3], this paper proposes an event-driven data collection mechanism. The events of an ILS monitoring system are divided into two types: hard real-time events and soft real-time events. Thus, the proposed system architecture ensures timeliness during data collection. Secondly, this paper provides a service-replacement fault-tolerant mechanism to address the challenge in fault tolerance control.

The rest of the paper is organized as follows. Section 2 discusses the related work on wireless sensor networks, smart homes, and smart spaces. Section 3 presents a real-time data collection system architecture including four steps: date generation, data transposition, data processing, and services. Section 4 describes a service-replacement-based timestamp matching algorithm to ensure the robustness and maintainability of the ILS monitoring system. Section 5 presents a prototype system that has been developed to validate the proposed approach. Section 6 concludes the paper and discusses future work.

II. RELATED WORK

A wireless sensor network (WSN) consists of nodes and devices which can process signal/data/information and communicate with each other or with computers, and interact with other systems and environments [6]. A comprehensive literature survey on WSNs can be found in [7].

ILS monitoring is a concept related to pervasive computing [8]. One of the representative projects in this area is the Gator-Tech Smart House initiative developed by the Mobile and Pervasive Computing Laboratory at the University of Florida [1]. Here, the Smart Home is a programmable pervasive space designed to assist the elderly in their daily lives to make them more comfortable and safer. Similar work has also been reported by Intille et al. [9]. Mills [10] proposed an Airjava technique which combines Java Jini with pico-cellular wireless technology to empower small devices to discover each other, to exchange programs, and to interact. The Airjava technique provides a practical way to address both the networking challenges and the human-information interaction issues of Smart Spaces. Similarly, Callaghan etc. [8] presented their understanding of the smart home in terms of interactions, which analyses socio-technical interaction and cyclic interaction. They believe interactions enable smart homes to

coordinate actions based on WSN. Janjic [11] argued that smart space automation can be seen as a part of the smart grid. Kainulainen [12] presented an interesting goal-based approach in defining smart home artificial intelligence. Cook et al. [13] studied how the inhabitants live in order to maximize their comfort and productivity by automating and predicting tasks in the house. Frans et al. [14] analyzed the usual algorithms of reasoning in a smart space, such as the Active-LeZi approach [15], the goal-based approach [16], and so on.

III. EVENT-DRIVEN REAL-TIME DATA COLLECTION

A. Data Transmission Cost Analysis

In a WSN-based ILS monitoring system, the processing of real-time data always shows a key contradiction that limited resources cannot meet the long-term running requirements of the system [17]. Wireless Sensors used by an ILS monitoring system, consisting of small devices, collect information through cooperating with each other. These small wireless devices (motes) consist of: microprocessor, memory, battery (for energy) and transceiver (for receiving and sending signals or data from one mote or to another) [17]. These motes are tiny, due to the constraints of environment and price, which limit the amount of the mote-owned resources. The cost of these devices depends on the parameters, like memory size, processing speed and battery, as described in [17, 18]. It can be computed by the following formula:

$$Cost = \sum_{i=1}^n \int_0^T nodeCost_i(t) dt \quad (1)$$

Inside this formula, the cost is the consumption of all nodes of a WSN during time period $[0, T]$, and n is the number of the nodes. $nodeCost_i(t)$ is the cost of node i at time t . Since $nodeCost_i(t)$ can be characterized by the sequence of vectors

$$nodeCost_i(t) = \left\{ \begin{array}{l} computingTime(t), energy(t), memory(t), \\ channelHolde(t) \end{array} \right\} \quad (2)$$

where $computingTime(t)$ is the data processing time of the microprocessor, and $energy(t)$ includes microprocessor energy consumption and signal transportation energy consumption; $memory(t)$ represents the usage of memory. $channelHolde(t)$ is used to reflect usage ratio.

Therefore, in order to meet the requirements of low power consumption and small memory size in WSN, the pretreatment of data is needed prior to transmission. This is because the primary energy is consumed during the data transfer process. Various algorithms and techniques have been proposed and developed to reduce or compress data transportation. Therefore, the objective of system optimization is to minimize the utilization of resources and transfer the maximum amount of data under the constraint of low power consumption. This makes the WSN data transmission a unique characteristic: data transmission is an event trigger.

B. Event-driven Technique Basics

An ILS monitoring system crosses three levels: the infrastructure layer, application layer, and service layer. This enables the event-driven mechanism to have its inherent

characteristics. The features of events in the infrastructure layer are different from those in the service layer.

Firstly, in the infrastructure layer, the sensor motes are running in an event-driven model. An internal operation of a sensor is triggered by one or more events. In embedded systems, the response of an event may be achieved using interrupts instead of a constantly running main loop. The transmission between events includes some data or a trigger signal. Secondly, the event-driven signaling mechanism is further configured to trigger the application layer. The data of events is transmitted in the server sockets [19]. Several terms are defined as follows:

- *Event* is an original impetus to push system to run, which affects some state, operation, or data in the system. An event consists of an *Event head* and an *Event body*.
- *Event head* includes event ID, event name, event timestamp, event occurrence number, and event creator.
- *Event body* includes state, event operation, data, and event object, where event operation at the service layer usually is a service invocation. The data transmission starts at the event generation, and ends at the finish of the related event's response.

C. Event-driven Data Collection Mechanism

Based on the event-driven architecture, an event-driven data collection mechanism is proposed to model a WSN-based ILS monitoring system. The event-driven architecture's inherent characteristics are extremely loose coupling and high discreteness [20, 21]. The event creator does not need to know what happens after the event is triggered. In the same way, the event receiver has no idea about how the event is created. Therefore, this mechanism is best used for asynchronous systems. Meanwhile, the framework used in our project also references the framework of Signal Processing in Node Environment (SPINE) [22]. In the remaining part of this section, an event-driven mechanism is analyzed in detail.

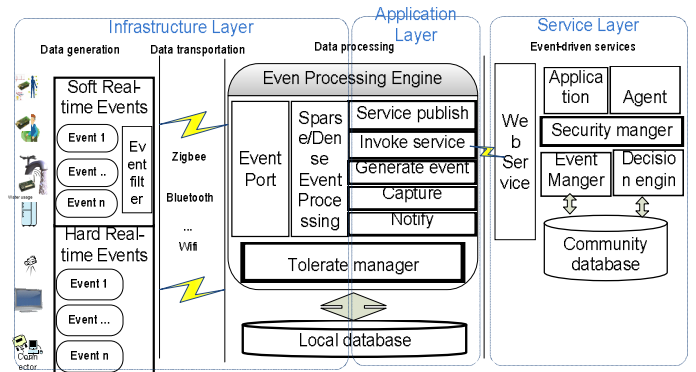


Figure 1. Event-Driven Real-time data collection diagram

An event-driven, real-time data collection mechanism consists of four steps: data generation, data transportation, data processing, and event-driven services, as shown in Figure 1. Since all data is encapsulated in events, the event flow is the

same as the data flow. An event starts with the environment triggering for the sensor, and culminates in its operating service. The four logical steps are described as follows:

- **Data generation:** In the hardware layer, all the events are triggered by the monitoring objects. For example, a water-using operation activates the sensor mote attached to the water-tap, which is a soft real-time event. The event includes an event head and a body, which maintain all related information. Due to the nature of the multi-hop network, the same event may arrive at the receiver several times. Therefore, an event filter is used to filter useless events. For example, the water is turned on and off twice and a second does not make sense. In order to make the system more stable and reliable, the event filter is very important.
- **Data transportation:** This step focuses on how an event is transported to the layer of data processing. Based on the different hardware techniques, event transport channels have corresponding protocols. Common transport protocols include ZigBee, WiFi, and Bluetooth [19]. In a practical monitoring environment, some smart home appliances also have their proprietary protocols.
- **Data processing:** The event has been transmitted to this step to be processed. Although smart sensor motes have some computing capabilities, their capabilities are very limited. Where some complex events happen, it is difficult for the motes to perform complex processing. Through the event port, different events in corresponding message styles are submitted to an event processing engine. The event processing rules and actions are defined in accordance to the needs of the interested parties, not of the event generators. Event processing creates a series of follow-up operations, i.e., service publishing, service invocation (from local mechanism or web mechanism), event generation, capture, and notification. All of these operations happen under fault-tolerant management.

A local database records three types of results. The first type is the received events which are used to ensure data security and integrity. The local database stores the collected data, which are the most complete first-hand records. In other words, the original data of the whole system are stored in every distributed local machine. Second are the operations records. For example, after the event generation is finished, the local database should add a record about the created event. The third type of record concerns the fault-tolerant control. Once the system encounters any exception or unfinished operation, the local database records the related information to ensure the robustness of the system.

It is important to point out that the operations of the data processing step are distributed into two layers in terms of the system structure, namely, the infrastructure layer and application layer.

- **Event-driven Service:** In this step, some events may initiate numerous subsequent activities, some of which may need to work collaboratively through the Internet. For example, when a high blood glucose event happens, the subsequent task is to invoke a phone vendor system to make a call for an ambulance or create a reminder. Namely, an event happens, and the related service is triggered.

In addition, events are not only generated at the infrastructure layer or in the data generation step. Event generation can occur at any layer and in any step. However, the lifecycle of an event will experience the same four steps as mentioned above. In the lifecycle of an event, the related data is transmitted step by step, flowing through different stages. In the following section, the discussion on fault tolerance will clearly illustrate this point.

IV. FAULT-TOLERANT MANAGEMENT MECHANISM

Although some fault-tolerant techniques of Web Services technology has been widely used to build low-cost, highly reliable, fault-tolerant systems, there is no special fault-tolerant technology that is fully available to meet the requirements of an ILS monitoring system which combines hardware and software in a wireless environment. As an ILS monitoring system is a highly distributed system that integrates infrastructure (device layer), middleware, and web services, normal fault-tolerant control algorithms of the Web services layer have difficulty managing the fault sensors located at the infrastructure layer. This fault management design requires a seamless connection and loose coupling between layers. This proposed mechanism is a timestamp matching algorithm based on service-replacement [23]. Figure 2 shows the principle of the proposed mechanism.

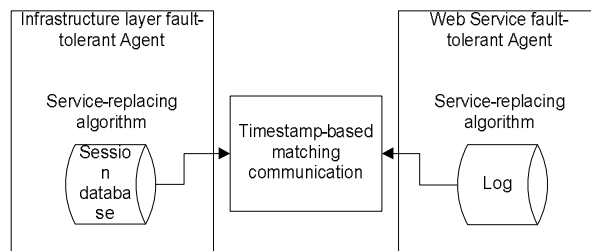


Figure 2. Service-replacement based timestamp matching

Since an event-driven architecture's inherent characteristics are loose coupling and high discreteness, and the data transportation and data operation are driven by events, it is difficult for the top layer to find a fault of the low layer. In order to solve this problem, timestamp matching is introduced to the fault-tolerant management algorithm. Service-replacement is independently running in both layers which detect the faults of each other through matching the timestamp of the records of the databases in the both layers. Figure 3 shows the fault-tolerant structure of the whole system.

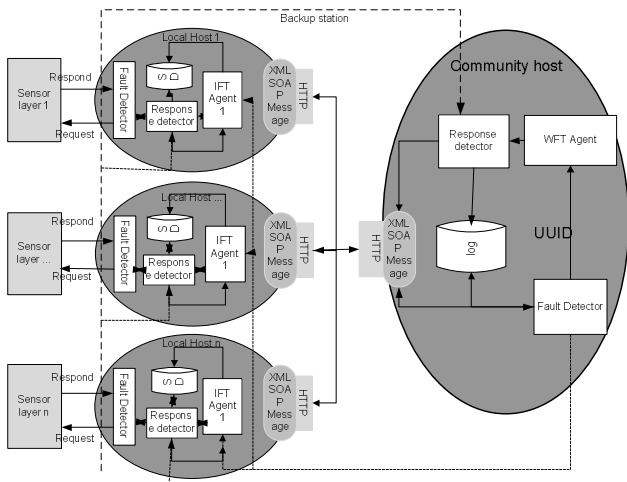


Figure 3. Service-replacement based timestamp matching algorithm structure

In terms of the function structure, this fault-tolerant algorithm can be divided into three modules: agent-based service management (FT Agent), fault detector-based management, and timestamp matching-based log management.

- **Agent-based service management:** a fault-tolerant agent is defined in this module, which collects the state of other hosts through a fault detector or reading records from the log database. Fault-tolerant agents (FT Agent) of both layers have the same function. Different fault-tolerant agents follow different rules in the management of the log data records.
- **Fault detector-based management:** this function module is in charge of fault detecting, fault notifying and fault record saving. When a fault event of a host is detected by a fault detector, it will be stored in a database. Then the fault-tolerant agent will get the UUID of the fault host from the fault detector's related report.
- **Timestamp matching-based log management:** The function of this module is to compare corresponding records in the databases of two hosts by the timestamp to determine the location of the breakpoint.

The fault-tolerant management design in an ILS monitoring system has two purposes: detecting system states and handling fault events. Thus, the system is kept running in a smooth, stable, and reliable state when some elements are added or removed. The fault detector generates a testing event in a certain frequency to check other hosts' states, and the states of all the services supplied by those hosts. Between the local machine and the related sensor layer, it is the local host that sends a message at a certain frequency to check whether the connected sensors are working. According to the response, the fault agent takes different actions. For example, if a normal acknowledgement message from the tested object is received,

this state information will be parsed from the event message, and saved into a service list in the local machine. If a time-out event message is received, all the related services will be set to a fault state. The processing of fault events is similar to a service replacement algorithm mentioned above. The details of the service-replacement are shown in Figure 4.

```

monitorFaultProcess(message){
    // message is a event message which is defined with wsdl
    //this function is to show a logical process.
    if(message=null) { //if received message is null,
        //fault detector continue monitor;
        continue monitor;
    }
    else if(message=faultEvent){
        // message is fault-event message.
        parse message and save the value into database;
        update serviceReplaceList;
        send_Backup_message;// the fault into other host;
    }
    else if(message=ACK_of_send_Backup_message){
        // message is a acknowledge of the backup operation
        update fault_detector;
        invoke service_recovery_management;//recovery operation
        // based on timestamp mapping;
    }
    else if(message=Service_Active_message){
        //it is event about the a service from fault status
        //became into available status.
        update fault_detector;
        send_Backup_message;// the fault into other host;
    }
    continue monitor;
}

```

Figure 4. The description of service replacement fault-tolerant algorithm

V. PROTOTYPE IMPLEMENTATION

The health of the elderly needs more care and attention. Most seniors like to live independently. To meet those needs, we have developed an Independent Living Space Monitoring prototype system to monitor the living space environment, health conditions and activities of senior residents in real-time [3]. The data items being monitored include: room temperature, motion, water usage, and fridge usage.

In the infrastructure layer, the monitoring data of room temperature, water usage, body motion, fridge door state, and door state are collected through the sensors embedded in motes, then sent to a base station mote which is connected to a local host machine (a computer). Figure 5 shows the data reading from the base station mote.

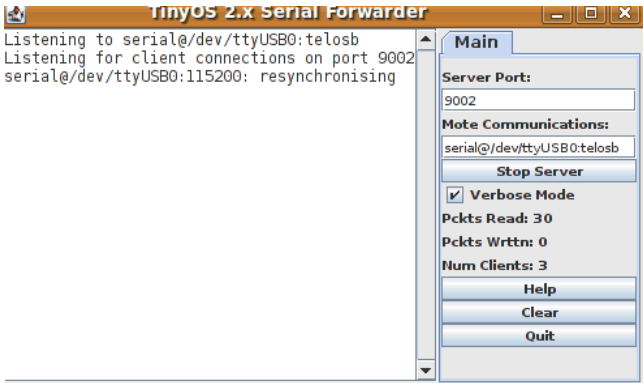


Figure 5. Data collection from the infrastructure layer

Figure 6 shows the implementation of the fault-tolerance mechanism. When the monitored environment is changed, an event is created by the related sensor. The event will be sent to the local host. After the event engine processes the event, the state of the event will be reported to all the related hosts. The fault detector will detect the state changes. When these fault events require human attention, the event processing engine will send notifications to the related users.

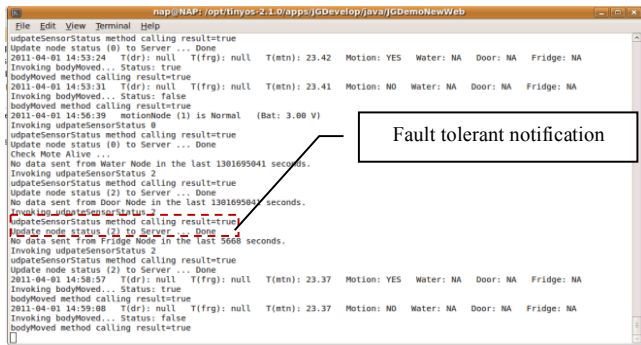


Figure 6. Fault tolerant report

A decision-making engine has been developed based on some general medical and health knowledge, as described in [24]. For example, when blood glucose level goes beyond a normal range, the system automatically alerts the caregivers. In our system, we have designed five levels of warning situations: normal, cautions, alert, serious, and dangerous. The end users are also divided into three categories: informal caregiver, formal caregiver, and first responder. The notifications will be delivered by text messages and e-mails to end users based on their roles.

The monitoring system adopts Telosb [25] motes provided by Crossbow Company as the communication nodes and base station node using the ZigBee communication protocol. Figure 7 shows a snapshot of the Web-based user interfaces.

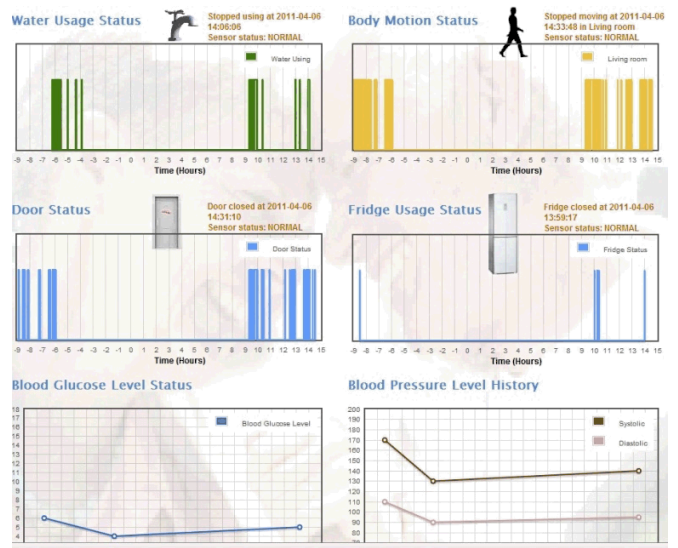


Figure 7. The Published Services of ILSMS

VI. CONCLUSION

A wireless sensor network for independent living spaces has been developed and applied to the monitoring of the physical environment of a living space and the health conditions of the elderly in order to ensure they can live independently and safely in their own homes. This paper presents a proposed WSN-based real-time data collection approach for independent living spaces. A service-replacement-based timestamp matching algorithm is developed to enhance the robustness of the system. Finally, a prototype system is implemented to validate the proposed approach.

This paper reports some preliminary results of a long term research initiative. Significant effort is still required to ensure the stability, reliability, and security of WSN-based real-time data collection for independent living spaces.

REFERENCES

- [1] S. Helal, W. Mann, Hicham, El-Zabadian, J. King, Y. Kaddoura and E. Jansen, "The Gator Tech Smart House: A Programmable Pervasive Space," the IEEE Computer Society, 0018-9162/05, 2005, pp. 64-74.
- [2] M.V. Ramesh, "Real-time Wireless Sensor Network for Landslide Detection," Proceedings of the Third International Conference on Sensor Technologies and Applications, 2009, pp. 405-409.
- [3] W. Shen, Y. Xue, Q. Hao, H. Xue, and F. Yang, "A Service-Oriented System Integration Framework for Community-Based Independent Living Spaces," Proceedings of IEEE SMC 2011, Anchorage, Alaska.
- [4] A. Stone, S. Shiffman, A. Atienza, L. Nebling, The Science of Real-Time Data Capture, New York: Oxford University, 2007.
- [5] D. Redelmeier, and D. Kahneman, "Patients' memories of pain medical treatments: Real-time and retrospective evaluations of two minimally invasive procedures," Pain, 1996, 66(1), pp.3-8.
- [6] Business process Management, [url] <http://www.eccentex.com/platform/workflow.html>.
- [7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey, Computer Networks," International Journal of Computer and Telecommunications Networking, 2002, 38(4), pp.393-422.

- [8] V. Callaghan, V. Zamudio, and J. S.Y.Chin, "Understanding interactions in the smart home," *PerAda Magazine*, 2011, doi: 10.2417/2200904.1625.
- [9] S. Intille, K. Larson, E. Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," *Pervasive Computing* 2006(4), pp.349-365.
- [10] K.L. Mill, *Proceedings of the Embedded Systems Workshop*, Cambridge, Massachusetts, USA, March 29–31, <http://www.usenix.org>.
- [11] D. Janjic, "Smart Home automation as part of the Smart Grid," *WFES* 2010, January 21, 2010. [url]: <http://ebookbrowse.com/beyond-smart-home-automation-wiser-home-pdf-d75816223>.
- [12] L. Kainulainen, *Reasoning in a Smart Home*, *AIOME* 2006, pp. 333-347.
- [13] D.J. Cook, M. Huber, K. Gopalratnam and M. Youngblood, "Learning to Control a Smart Home Environment," *Innovative Applications of Artificial Intelligence*, 2003, [url]: <http://ranger.uta.edu/~holder/courses/cse6362/pubs/Cook03.pdf>.
- [14] F.T. Vadén and I. Koskinen, "Introduction: Living in Metamorphosis - the Whys and Hows of Proactive Home Design Research," *The Metamorphosis of Home*, Frans, ja Ilpo Koskinen (ed.), Tampereen yliopisto, 2005, pp.7-27.
- [15] K. Gopalratnam and D. J. Cook, "Active LeZi, An Incremental Parsing Algorithm for Sequential Prediction," *FLAIRS* 2003, American Association for Artificial Intelligence (www.aaai.org). 2003, pp. 38-42.
- [16] T. Heider and T. Kirste, "Multimodal appliance cooperation based on explicit goals: concepts & potentials," *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, 2005, pp. 271-276.
- [17] D. Kumar, T.C. Aseri, and R.B. Patel, "Analysis on Deployment Cost and Network Performance for Heterogeneous Wireless Sensor Networks," *International Journal of Computer science & Information Technology (IJCSIT)*, 2009, 1(2), pp. 109-120.
- [18] R. Kay and F. Mattern, *The Design Space of Wireless Sensor Networks*, *IEEE Wireless Communications*, 2004, 11(6), pp.54-61.
- [19] P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, Y.F. Hu "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer Communications*, 2007, 30(7), pp.1655-1695.
- [20] B.M. Michelson, *Event-Driven Architecture Overview*, Boston, MA 02109, 2006(2), <http://dx.doi.org/10.1571/bda2-2-06cc>.
- [21] B. Klievink, M. Janssen, M. Lankhorst, and D. V. Leeuwen, "An event-Driven Service-Oriented Architecture for Coordinating Flexible Public Service Networks," *Service Architectures and Interoperability*, 2008, pp. 133-140.
- [22] R. Giannantonio, F. Bellifemine, M. Sgroi, "SPINE (Signal Processing in Node Environment): a Framework for healthcare monitoring applications based on Body Sensor Networks," *Telecom Italia Lab, White Paper*, <http://spine.tilab.com/papers/2007/WhitePaper.pdf>.
- [23] F. Lahrach, A. Abdaoui, A. Doumar, and E. Chatelet, "A novel SRAM-based FPGA architecture for defect and fault tolerance of configurable logic blocks", *2010 IEEE 13th International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2010, pp. 305-308.
- [24] E. Hendricks "Evangelical Offers May Blood Pressure and Blood Sugar Clinics", http://www.northcentralpa.com/news/2011-04-13_evangelical-offers-may-blood-pressure-and-blood-sugar-clinics.
- [25] Willow Technologies Limited, "Low Power Wireless Sensor Module, TelosB", http://www.willow.co.uk/html/telosb_mote_platform.html.