Sustainable Sensor based Environmental Information Systems for Smart Cities

Ruthbetha Kateule, and Andreas Winter
Carl von Ossietzky University Oldenburg, Oldenburg,
Germany Department of Computer Science
kateule@se.uni-oldenburg.de, winter@se.uni-oldenburg.de

Abstract

Smart cities use sensor based environmental information systems (SEISs) to promote effective utilization of the available resources and better services to citizens. SEISs require continuous and long-term measurements of environmental parameters such as temperature, humidity, gases, etc. However, the increasing inevitable changes in stakeholder goals and requirements of systems during evolution affect the energy efficiency of SEISs. As a first step towards sustainable SEISs, this work proposes an architectural design to analyse energy and resource consumption of SEISs in order to detect parts of the system with higher consumption of energy and resources. So as an engineer will be able to use the information obtained from such analysis together with energy efficient techniques to improve the energy efficiency of SEISs. This work extends the ongoing research of designing a reference architecture for SEISs with sustainability perspective.

Keywords: Smart City, Sensor based Environmental Information Systems (SEISs), Software Architecture, Technical Sustainability, Energy Efficiency.

1 Introduction

Smart cities use sensor based environmental information system (SEISs) to monitor and control environmental events in order to e.g. reduce traffic congestion, improve the quality of water, soil and air for the reduction of pollution and promotion of healthy environment [1, 2]. Such systems manage the data about soil, water, air and other objects revolving around the city. SEISs cover a wide range of aspects that contribute to the goals of environmental monitoring [3]. SEISs require continuous and long-term measurements of environmental parameters such as temperature, humidity, gases, etc. However, the increasing inevitable changes in stakeholder goals and requirements of the system and the fact that the quality of systems degrades during evolution make the maintenance and evolution of SEISs to be necessary and hard. Such situation is caused by several factors including wrong design decisions, bad coding and others which led to technical debt [4]. This is reflected through an unexpected increase in consumption of energy and resources i.e. CPU, memory, bandwidth, etc. which in turn affect negatively the performance and energy efficiency of the system [5].

Software systems are regarded to be sustainable, if they are energy-efficient with low environmental impacts and affect social and economic sustainability positively [6]. Hence the sustainable SEISs can be achieved primarily by improving power consumption. This could be initiated in the design phase during the development of software architecture [7]. Software architecture has been acknowledged by many researches as the base ground or foundation of any software system including design guiding principles and evolution [8, 9, 10]. As such, stakeholders can reason about their needs or concerns and main software quality attributes i.e. maintainability, extensibility, scalability, etc. A software architecture provides a pro-active planning towards the sustainable system. From an architecture point of view, sustainability of a system is determined in terms of resources and energy consumption with respect to maintenance and evolution [6]. Additionally, reference architecture represents a generic template software architecture which facilitate rapid development of concrete architectures by reusing architecture knowledge [11]. Hence reference architecture is required to facilitate the rapid development SEISs.

In response to the previous efforts of identifying the essential viewpoints of the reference architecture for SEISs [3], this work realizes the need of architectural design with sustainability perspective. Two important concepts are highlighted: (1) Meaning of a sustainable software (2) How to design a sustainable SEISs? This work considers *Technical Sustainability* as the principal dimension of sustainability due to its central objective towards a long-time usage of software system. Technical sustainability becomes a critical concern during the design phase [12] and is expressed in terms of energy efficiency of a system which include both energy and resources consumption.

The remainder of this paper is organized as follows. The foundation concepts are provided in Section 2. The proposed architectural design for sustainable SEISs is provided in section 3. Section 4 presents the applicability of the proposed architectural design. The paper ends with the conclusion in Section 5.

2. Foundations

This section presents the basic concepts that the proposed approach in this work is built upon. Section 2.1 outlines the definition and various dimensions of software sustainability. Software architecture is discussed in Section 2.2. Section 2.3 discusses the existing approaches that analyzes energy consumption of software at the architectural level.

2.1. Sustainability

Sustainability is a generic term used widely with respect to a particular context. For instance, sustainability refers to the ability of harvesting or using a resource in such a way that it is not depleted or permanently damaged [6]. While in another context, sustainability is defined as the ability to meet the current needs without compromising the future demands [20]. According to the UN [20], the development of the sustainable system is required to satisfy the following three dimensions or pillars as shown in Figure 1 which demonstrates common visualization of sustainability in terms of three separate concerns [219+9] together with the usage of SEISs in smart cities:

- **Economic sustainability** ensures the growth of the economy based on the ecosystem by the integration of both environmental and social concerns into a business. To be economically sustainable, a software should yield a long-term profit over low running costs [20]. SEISs are economically sustainable, if the cost of development, operations and maintenance are reasonable for instance advancement in technology reduced the costs of sensor nodes forming wireless sensor networks (WSN), eventually makes SEISs be cost-effective [19]. This could also be facilitated through adoption of suitable design choices such as use of open source components and architectural design patterns [7].
- **Environmental sustainability** focuses on the development of software systems with minimal environmental impacts [20]. This refers to the improvement of human welfare by conserving natural resources. SEISs specialize in monitoring and controlling environmental parameters to ensure the better quality of environmental resources such as water, air, and soil [1]. SEISs tend to reduce negative impacts of environmental phenomena so as to conserve the environment and biodiversity in the city.
- **Social sustainability** promotes social equity and ethical consumerism. The software systems are required to foster and maintain better social added values e.g. raise of capital [2]. SEISs seek to minimize undesirable impacts of environmental events such as fire, flood, air pollution and others in order to establish a city with better socio-conditions i.e. healthy people (and other living organisms) and environment [4].

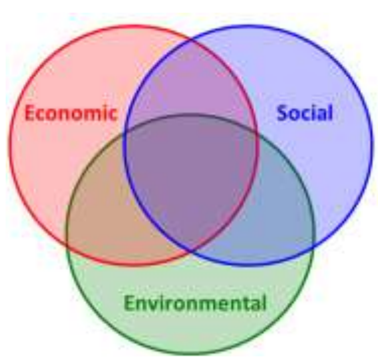

Figure 1: Dimensions of Sustainability [22]

According to that definition of sustainability provided in [6]. A system with long lifetime and efficient utilization of resources are main concepts underpinning sustainability. Hence the inappropriate consumption of energy and resources could compromise the sustainability of SEISs. However, the fundamental principle regarding how to design long-lasting SEISs with efficient utilization of resources is missing. Thus, another aspect (dimension) of sustainability mainly *Technical Sustainability* should be considered. Technical sustainability refers to the long-time usage of systems (information systems) and their adequate evolution with respect to the changing surrounding conditions and requirements [2]. With the regard to the specified definition of sustainability, technical sustainability is reflected in terms of the energy-efficient system since such system runs for a long-time with efficient utilization of resources [6]. The concerns of technical sustainability could be satisfied with less effort if energy efficiency is considered as critical concern explicitly from the early phase of software development (during the design phase of software architecture). Since the architecture is the foundation of any software and hence determines the quality of a system [9].

2.2. Software Architecture

Software architecture presents structure or structures of the system which is made up of components (both hardware and software), their relationships to each other and to the environment [6]. Additionally, software architecture includes the principles guiding the designs and evolution of such system [8]. Reference architecture is a generic template software architecture that preserves the established solutions or knowledge to assist the design, description, and analysis of concrete architectures by reusing architecture knowledge [11]. This work is an extension of the on-going research for designing reference architecture for SEISs with sustainability perspective. The construction of software architecture influences strongly the quality of resulted system i.e. the degree to which the specified architecture satisfies the requirements determines the quality of such software [13]. Software systems are highly dependent on their specified architecture for efficient and effective maintenance and evolution [7]. A desirable approach towards energy efficient SEISs is established by analyzing and optimizing the consumption of energy and resources at the architectural level.

2.3. Existing Energy Efficient Approaches

There are several software engineering approaches which support the analysis of the consumption of energy and resources of software systems. For instance, an energy consumption analysis for specific architectural styles is proposed in [15]. The measurements on components and connectors as the architectural elements resource were performed according to the related architectural styles e.g. Remote client energy cost metric estimates the energy consumption of client connector based on sending requests and receiving response [15].

Simon et al. proposed energy profiler named Eprof [13]. Eprof determines the hot spots or sections of smartphone applications which consume energy. Such profiler measures power consumer activities of an application and then maps to the program entities e.g. methods which caused them.

Tan et al. used macro-models to analyze the energy consumption of embedded applications [14]. In which, a Software Architecture Graph (SAG) is used to design software architecture. The SAG is made up of nodes for the tacks and directed edges for relationships between the components in the graph. The energy consumption of the architecture is estimated by macro-models for the tasks and edges after the generation of SAG. SAG represents behavior of the system during software execution.

Although there are several solutions towards energy efficient systems, most of them are designated for specific domains apart from SEISs and little work focused in addressing energy efficiency of software on the architectural level. The main goal of this work is to propose an architectural design that will utilize energy profiling to optimize energy efficiency of SEISs.

3 Proposed Architectural Design

The primary goal of SEIS is to monitor and control environmental events via sensors and actuators respectively. The main functionalities of such systems are data acquisition from the environment via sensors, data analysis, processing and reporting (viewing) via a server and controlling of the environmental events through actuators. To achieve energy efficiency of SEISs, Section 3.1 discusses the actual needs of the approach and then Section 3.2 describes the proposed design.

3.1. Requirements for Energy Efficient SEISs

Energy is one of the critical resources in SEISs especially for SEISs which employ wireless sensor networks (WSNs). Since WSNs are made up of sensor nodes which are low-cost devices with limited power (small battery). Energy efficiency becomes a critical concern when such sensors are deployed for large-scale environmental monitoring without human control or intervention and absence of energy efficient techniques [4, 1]. Additionally, the power consumption of servers depends on the load. The raised requirements of the clients during evolution may led to the deployment of more software components or serving more clients. This induces more loads to the server increases energy and resources consumption which eventually increases running and maintenance costs.

Although physical processes located in hardware consume energy and other available resources, the software causes inefficient use of hardware which led to wastage of energy and resources. Since the software controls the hardware, the wastage of energy and resources cannot be solved by relying on only energy and resource efficient hardware. But also, energy and resource efficient software is required. Therefore, there is a need for energy and resources transparency in software design. Software architecture influences the power consumption of the system by imposing design decisions in the distribution of components (and loads), communication styles and cost assessment of expected power consumption in the system [16]. To improve the sustainability of SEISs, the influence of architectural design on energy and resource consumption should be considered. To achieve this, an energy efficiency viewpoint is required to identify the energy hotspots, analyse energy consumption and provide various alternatives / mechanisms to optimize energy efficiency of SEISs.

3.2. Proposed Architectural Viewpoint

This section proposes a generic architectural design (viewpoint) with strong emphasize on separation of concerns and modularity to fulfill the requirements identified in Section 3.1. Such proposed design intends to conserve the energy consumption of SEISs.

The proposed viewpoint is adopted from the execution viewpoint of Siemen's View Model [17] with an extension of *Energy Monitor and Optimize* and is shown in Figure 2. This viewpoint describes the how the functionalities of SEISs' are distributed and communicate interms runtime platform elements i.e. processes, threads etc., (through *Communication Mechanism, Communication Path, Platform Element,* and *Runtime Entites*) and also how the physical resources (*Hardware Resource*) are allocated to them. The *Energy Monitor* observes the activities performed by physical resources (*Hardware Resource*) for instance for sensors are CPU, Memory and Transceiver. Then *Energy Monitor* records code location of such activities and estimates the energy consumption of each physical resources via energy model (EModel) to identify the potential energy hotspots. Since each *Hardware Resource* consists of energy model (*EModel*). Afterwards, *Energy Monitor* utilizes *EAnalyser* to determine if the consumed energy is above the threshold or will affect the overall performance of the system.

To improve the consumption of energy the *Optimizer* is used to execute alternative solutions in energy hotspots by reengineering techniques interms of new algorithms relating to data storage, processing and transmission algorithms. This provides a clear understanding of which source code lines or blocks consume more energy or resources during runtime. Hence an engineer can employ various techniques in energy hotspots to improve energy efficiency of system. For instance, Refactoring restructures the system by removing energy code smells which are energy-wasteful parts of the system i.e. Binding resources too early and Third part advertising energy [18]. Self-adaptation utilizes network usage scenarios to measure energy costs of functionalities, then create various energy profiles for a set or subset of functionalities that conserve energy [19]. The *Hardware Resource* may have a controller (*DController*) which controls the operation of *Hardware Resource*.

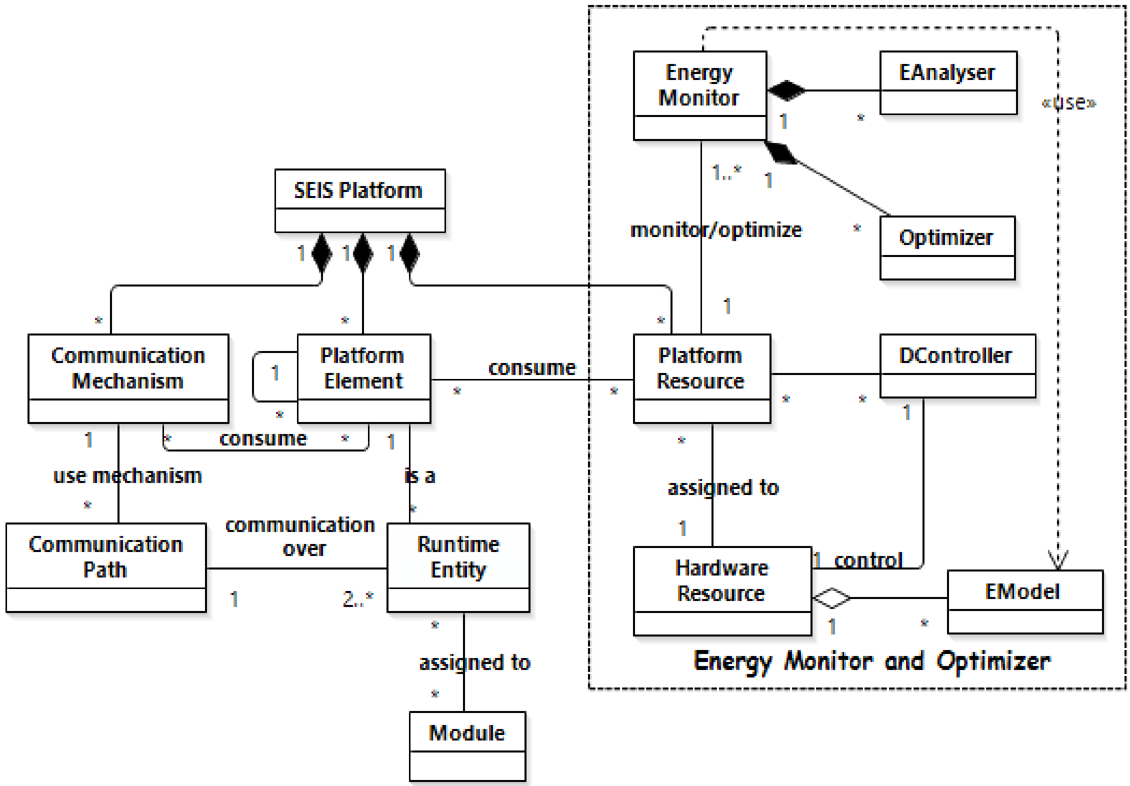

Figure 2: Execution Viewpoint of SEISs with Energy Consumption Perspective

4 Applicability of the Proposed Architectural Design

The proposed architectural viewpoint in Section 3 guides the construction of energy efficiency views for concrete SEISs based on stakeholders' criteria in the selection of execution scenarios. This view describes how a software system utilize the hardware resources during runtime. Taking air quality monitoring system as a smart city application which intends to monitor the quality of air in real time, making essential decisions, generating reports and informing the citizens in smart cities. Reports of air quality status in smart cities include warnings and forecasting. Air quality monitoring systems are usually made up of various sensors such as gases, humidity, temperature and other sensors. Such sensors form Wireless Sensor Networks (WSNs) and communicate with each other and with the gateway by using IEEE 802.15.4 (ZigBee) and the collected information is sent to the server by using General Packet Radio Service (GPRS) communication protocols. The server hosts such information systems by integrating the collected information, analyzing and disseminating reports to people through Global System for Mobile communication (GSM) short messages.

To ensure the sustainability of air quality management system, *Energy Monitor* (from the proposed viewpoint) is instantiated in the design view for improving the energy efficiency of such system during execution. *Energy Model* estimates the energy consumption of CPU, memory and transceiver of sensors to identify potential energy hotspots. The resulted estimations will have significant effects on the longevity of air quality management system. Since, an engineer is able to use such estimations to remove code smell, dead codes and rectify bad designs based on the adopted techniques i.e. refactoring and self-adaptation and hence led into energy efficient and sustainable air quality management system.

5 Conclusion and Future Work

Smart cities promote effective utilization of the available resources and better services to the citizens. The smart city uses sensor based environmental information systems (SEISs) to provide services such as water, land and air quality management. However, the increasing need for a long lifespan and continuous availability of SEISs makes the development of such systems to be difficult. This is due to inevitable changes in stakeholder concerns and other raised requirements of the system. It is very important to incorporate sustainability explicitly in the early phases of software development in order to avoid unnecessary wastage of efforts and time in developing a system with a shorter lifetime. Since software architecture promotes sustainability. Therefore, this work proposes architectural design for facilitating the development of sustainable SEISs.

This works presents the conceptual design with emphasize on separation of concerns and modularity to achieve energy efficient SEISs. With the proposed architectural design, developers can identify energy consumed hotspots and then employ various reengineering techniques to improve the energy efficiency of SEISs. The applicability of such proposed design is elaborated through air quality management system as a smart city scenario. In the future work, the actual implementation of the proposed architecture will be conducted.

References

[1] M. Penza, D. Suriano, M. G. Villani, L. Spinelle, and M. Gerboles, “Towards air quality indices in smart cities by calibrated low-cost sensors applied to networks, ” In IEEE SENSORS 2014 Proceedings, pp. 2012-2017, 2014.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. IEEE Internet of Things Journal, pp. 22-32, Feb 2014.

[3] R. Kateule and A. Winter, “Viewpoints for sensor based environmental information systems, ” In Environmental Informatics-Stability, Continuity, Innovation, 2016.

[4] M. Ilkiewicz and J. Letouzey, “Managing technical debt with the sqale method, IEEE Software, ” pp. 44-51, 11 2012.

[5] R. N. Taylor, N. Medvidovic, and E. M. Dashofy, “Software Architecture: Foundations, Theory, and Practice, ” Wiley Publishing, 2009.

[6] C. Calero, M. Angeles Moraga, and M. F. Bertoa, “Towards a software product sustainability model, ” CoRR, abs/1309.1640, 2013.

[7] C. Venters, R. Capilla, S. Betz, B. Penzenstadler, T. Crick, S. Crouch, E. Y. Nakagawa, C. Becker, and C. Carrillo, “Software sustainability: Research and practice from a software architecture viewpoint, ” Journal of Systems and Software, pp. 174-188, April 2018.

[8] ISO/IEC/IEEE, “Systems and software engineering - architecture description, ” ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), pp. 1 -46, 1 2011.

[9] S. Malek, C. Seo, S. Ravula, B. Petrus, and N. Medvidovic, “Reconceptualizing a family of heterogeneous embedded systems via explicit architectural support, ” 29th ICSE'07, pp. 591-601, 2007.

[10] N. Medvidovic and R. N. Taylor, “A classification and comparison framework for software architecture description languages, ” IEEE Trans. Software Engineering, pp. 70-93, January 2000.

[11] C. Hofmeister, R. Nord, and D. Soni, “Applied Software Architecture, ” Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.

[12] H. Koziolek, “Sustainability evaluation of software architectures: A systematic review,” In Proceedings of the Joint ACM SIGSOFT Conference, QoSA-ISARCS '11, pp. 3-12, 2011.

[13] S. Schubert, D. Kostic, W. Zwaenepoel, and K. G. Shin, “Profiling software for energy consumption,” In 2012 IEEE International Conference on Green Computing and Communications, pp. 515-522, Nov 2012.

[14] T. K. Tan, A. Raghunathan, and N. K. Jha, “Software Architectural Transformations,” pp. 467-484, Springer US, Boston, MA, 2003.

[15] C. Seo, S. Malek, and N. Medvidovic, “Estimating the energy consumption in pervasive java-based systems, ” In 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (Per-Com), pp. 243-247, March 2008.

[16] C. Stier, A. Koziolek, H. Groenda, and R. H. Reussner, “Model-based energy efficiency analysis of software architectures, ” ECSA, volume 9278 of Lecture Notes in Computer Science, pp. 221-238. Springer, 2015.

[17] D. Soni et al., “Software architecture in industrial applications, ”In International Conference on Software Engineering, pp. 196–207, 1995.

[18] M. Gottschalk, J. Jelschen, and A. Winter, “Energy-efficient code by refactoring, ” Softwaretechnik-Trends, 33(2), 2013.

[19] L. Ardito, G. Procaccianti, M. Torchiano, and A. Vetro, “Understanding green software development: A conceptual framework,” IT Professional, 17(1), pp.44-50, 2015.

[20] United Nations World Commission on Environment and Development. Report of the world commission on environment and development: our common future, 1987.

[21] A. D. Basiago. Economic, social, and environmental sustainability in development theory and urban planning practice. Environmentalist, 19(2), 1998.

[22] The Green Art Lab Alliance. Sustainability, 2018.