Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Especialidad en Sistemas Embebidos

Defining an air pollution sensor inside the car

TESIS que para obtener el **GRADO** de
ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presenta: **LUIS ARTURO GONZÁLEZ CASILLAS**

Tutor **HÉCTOR ANTONIO RIVAS SILVA**

Tlaquepaque, Jalisco. agosto de 2021.

Defining an air pollution sensor inside the car

Luis Arturo Gonzalez Casillas
Dept. Systems and Electronics
Instituto Tecnologico de Estudios Superiores de Occidente (ITESO)
Guadalajara, Mexico
lgonzalez.casillas@iteso.mx

Hector Antonio Rivas Silva
Dept. Systems and Electronics
Instituto Tecnologico de Estudios Superiores de Occidente (ITESO)
Guadalajara, Mexico
harivas@iteso.mx

***Abstract*— This paper provides the specification and definition of a system capable of reading pollution inside the cabin of an automobile and shows the measurements to all passengers. Requirements elicitation, system requirements analysis, system architectural design, software requirements analysis, software architectural design, software detailed design and unit construction phases from the automotive Spice process reference model will be explained in detail and some examples will be shown. This paper can function as a guide to define and specify a system using different methodologies between these phases. The system design proposed here can be a starting point for future applications related to automobile pollution measurements, for example, reading environmental pollution and monitoring automobile emissions.**

Keywords—air pollution, BDD, Automotive SPICE, AUTOSAR

I. Introduction

Nowadays, humanity is facing environmental and health threats due to air pollution. According to the World Health Organization, an estimated 4.2 million deaths per year worldwide are caused by air pollution [1].

Automobiles produce 60% to 70% of air pollution in the world [2], hence, the automobile industry is a major contributor. Several nations are pressing this industry to seek less polluting processes and products by creating regulations to reduce emissions from industrial sources, emissions from vehicles and engines through new stringent emission standards and cleaner-burning gasoline [3].

This paper proposes that the starting point to reduce air pollution is by making people aware of its impact. The main purpose of this project is to design a system capable of reading pollution inside the cabin of an automobile and making it visible to all passengers.

The processes requirements elicitation, system requirement analysis, system architectural design, software requirements analysis, software architectural design, software detailed design, and unit construction from the Automotive SPICE process reference *model* were performed and illustrated in this paper; examples and tips for the development of each phase are provided as well as details on the system proposed for the solution of the air pollution threat.

II. Methodology

The Automotive Spice process reference model was developed to fulfill the specific needs of the automotive industry and it complies with the requirements of ISO/IEC 33004. [4] The reference model SPICE group processes into three process categories: primary life cycle processes, organizational life cycle processes and supporting life cycle processes [4].

Fig. 1 shows the different processes categories and every process which conform it.

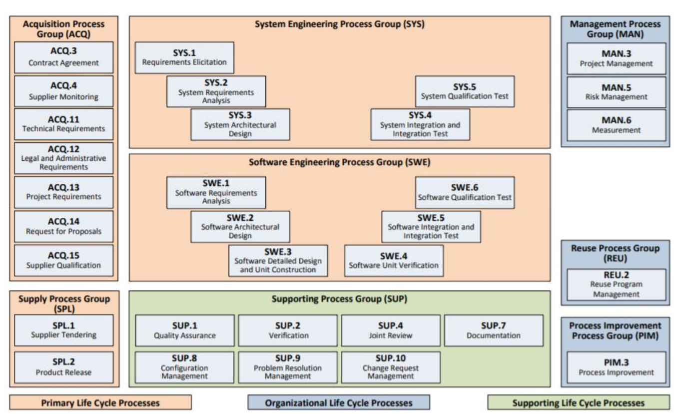

Fig. 1. Automotive SPICE process reference model, 2015-07-16 [4]

For the specification and definition of this system, processes SYS.1, SYS.2, SYS.3, SWE.1, SWE.2, and SWE.3 from the primary life cycle processes were followed.

A. *SYS.1 Requirements elicitation*

BDD (Behavior-Driven Development) method was selected to clearly understand stakeholder expectations. BDD aims to close the gap between technical understanding and stakeholder expectations through writing use case scenarios that are understandable to all involved. According to BDD, to write use case scenarios a natural language and a "Give-When-Then" structure shall be used, besides it should be focused on describing the behaviors of the system [5].

An example from some of the use cases for this system are as follows:

1. Given the sensor is inside the cabin of an automobile
When the automobile is switched on
Then ensure the sensor is powered on
And ensure air pollution is measured by the sensor

2. Given the automobile can move
When the sensor performed a reading
Then ensure GPS coordinates are bound and stored with the readings captured by the sensor

3. Given the system does not contain a screen
When the user wants to see the readings
Then ensure communication through Bluetooth is available
And ensure the sensor can be paired with the user's cellphone
And ensure the user can see the readings from the sensor through a Bluetooth terminal on the cellphone

Once all the use cases were documented, a revision with stakeholders was performed to acquire feedback. Multiple iterations occurred before receiving approval and continue with following processes.

B. *SYS.2 System requirements analysis*

To define system requirements, an overall understanding of expectations, constraints and physical/functional interfaces must be accomplished.

Requirements were written following the Expanded Guidance for NASA Systems Engineering [6], which states that requirements should be defined in acceptable "shall" expressions and must contain only one shall per statement. Each requirement shall be validated.

For this system, every system requirement was validated using the following questions proposed on the Expanded Guidance for NASA Systems Engineering:

- Are the requirements written correctly?
- Are the requirements technically correct?
- Do the requirements satisfy stakeholders?
- Are the requirements feasible?
- Are the requirements verifiable?
- Are the requirements redundant or over-specified?

Systems requirements provide a complete description and constraints on how the system shall react to a specific situation, as well as it also mentions what kind of communication protocols will be implemented to meet customer expectations and the assigned budget. System requirements mention what hardware elements will be used to generate the expected solution and it also delimits the boundaries for the software.

C. *SYS.3 System architectural design*

To create the system architecture a boundary diagram was created. Boundary diagrams show all the elements that interact within the system including their interfaces.

Fig. 2 shows all the elements that are part of the system and the interaction between them. A satisfactory system architecture shall consider all elements that interact with the system, internal and external. Issues at the integration, testing, and validation phases could be avoided by considering all elements that interact within the system.

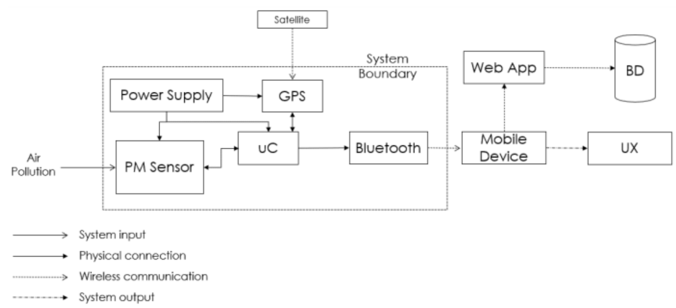

Fig. 2. System Architecture, Boundary Diagram

D. *SWE.1 Software requirements analysis*

Once system requirements and system architecture are completed, software requirements can be defined. Every system requirement shall be traced to at least one software requirement and software requirements shall consider and cover all the system architecture too. Software requirements and software architecture could be created at the same time to gain a better understanding of how the software modules will be distributed. What elements do the software need to cover all the functionality expected by the system? It is the question software requirements and software architecture shall response.

Creating sequence diagrams based on use case scenarios, system requirements and system architecture clarifies which software modules will be needed.

Fig. 3 shows an example of a sequence diagram created for the definition of the system. Software requirements were created having into consideration SOLID principles of object-oriented programming.

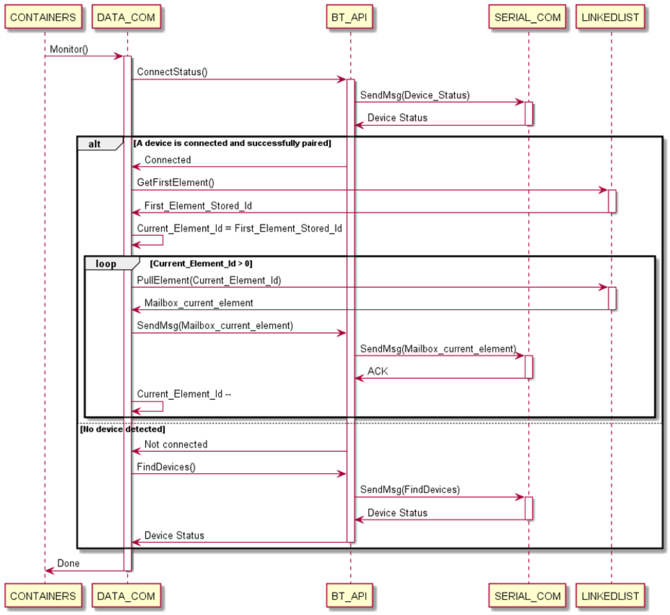

Fig. 3. Sequence Diagram – Sending Stored Data

Grouping software requirements into functionalities or modules become easier to read, understand and implement, therefore, software requirements were written by grouping requirements into modules, where each module has one and only one responsibility.

In addition to software requirements, a traceability record shall be created and documented in a way that a software requirement can be traced back to a system requirement and can be found in the system architecture, hence, software requirements shall also be traced to all software elements. Traceability must be possible backward as well as frontwards.

To facilitate software test specifications, software requirements were categorized into functional and non-functional requirements.

E. *SWE.2 Software architectural design*

AUTOSAR classic platform architecture was selected to define software architecture. AUTOSAR is a worldwide collaboration of multiple companies from the software, automotive, and semiconductor industry. Standardization of functional interfaces and basic functions are the primary goal [7].
All the elements in the software, the interaction between them, and the way they are located into different layers are shown by the software architecture. The software architecture for this system is shown in Fig. 4

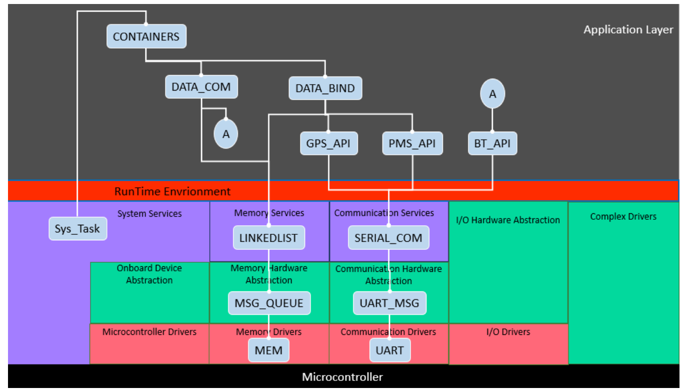

Fig. 4. Software Architecture using AUTOSAR layers

F. *SWE.3 Software detailed design and unit construction*

Software requirements, architecture and sequence diagrams shall be considered to create the software detailed design.

The functionality of an interface from a module can be visually represented by activity diagrams. Software detailed design could be documented by activity diagrams of each interface from every module.

Defining component interfaces shall be made by following the scope of each module defined on the software requirements and architecture. An interface shall do only one thing according to the SOLID principles.

Figure 5 shows an activity diagram for the interface called monitor of the module or component named DATA_COM.

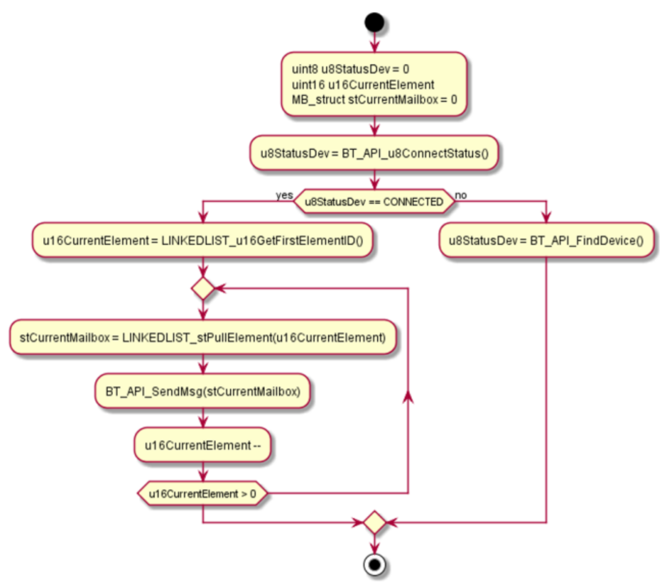

Fig. 5. Activity Diagram – Monitor interface from DATA_COM module

Unit construction of modules was developed after the creation of activity diagrams for each interface on every module.

An example of traceability can be seen in the implementation of the module DATA_COM. Justification of the activity diagram of Figure 5 can be traced to the sequence diagram of Fig. 3.

III. Conclusion

Air pollution is one of the causes of biggest threats nowadays like global warming and health issues. By the implementation of this system, the population would start being aware of this “invisible” issue by making it visible. Also, this work can be used as a guide to specify and define any type of system involving software and as a starting point for applications related to automobile pollution measurements.

Acnowledgments

This work was supported by the Mexican federal government through the CONACyT agency.

References

[1] Who.int. 2021. Ambient air pollution. [online] Available at: <https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/ambient-air-pollution> [Accessed 7 July 2021].

[2] Swami, Abhishek. (2018). Impact of Automobile Induced Air Pollution on roadside vegetation: A Review. ESSENCE International Journal for Environmental Rehabilitation and Conservation. 9. 101-116. 10.31786/09756272.18.9.1.113.

[3] US EPA. 2021. Reducing Emissions of Hazardous Air Pollutants | US EPA. [online] Available at: <https://www.epa.gov/haps/reducing-emissions-hazardous-air-pollutants> [Accessed 7 July 2021].

[4] 2016. Automotive SPICE. 3rd ed. [ebook] Available at: <http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf> [Accessed 7 July 2021].

[5] D. North, “Introducing BDD,” Better Software Magazine, 2006.

[6] 2016. Expanded Guidance for NASA Systems Engineering. Washington, D.C: National Aeronautics and Space Administration, pp.92 - 107.

[7] cooperation, A., 2021. AUTOSAR. [online] Available at: <https://www.autosar.org/> [Accessed 9 July 2021].