We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

5,300
Open access books available

130,000
International authors and editors

155M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

System Engineering Method for System Design

Guillaume Auriol, Claude Baron,
Vikas Shukla and Jean-Yves Fourniols
CNRS, LAAS,
Université de Toulouse, UPS, INSA, INP, ISA, UT1, UTM, LAAS
France

1. Introduction

The purpose of this chapter is to present some educational materials, the process and the outcomes to teach an engineering approach applied to a practical development case. The starting point is the requirements of an application of remote supervision of a room with several parameters: light, temperature and movement (intrusion into the room or movement of an object within the room). This application is based on wireless terminal nodes composed of a sensor, a microcontroller and a telecommunication module. Several rooms can be interconnected, so it must be possible to use the sensors of each room of a given site simultaneously. Various issues can be raised during teaching on wireless sensor networks (Kotzl & Essien, 2005): electronic design, risks to humans (Brownsell et al., 1999), energy management, telecommunication technologies, etc.

During the course, students have to learn and apply a 'systems engineering' (Ullrich K.T. and Eppinger S.D, 2003), (Terry A. Bahill and Clarck Briggs, 2001) approach based on standards in the field (Martin, 1998), (ISO15288, 2008), (IEEE1220, 2005) to solve a problem with numerous design options. Several off-the-shelf software and hardware components are at the students' disposal: a panel of telecommunication modules, different communication and signalling protocols, etc. They start by studying the requirements to extract an exhaustive list of needs. They must then propose and evaluate functional and architectural solutions, and finally implement the chosen solution in order to validate their 'systems engineering' approach.

Section 2 gives an overview of the method to follow to design a telecom system. Section 3 depicts the application through stakeholder's needs. Sections 4 to 6 detail the four steps of the method with (4) definition of stakeholders' needs and definition of technical requirements, (5) design of functional and (6) physical architectures. Section 7 presents the component realization, the component integration and the system validation. Finally, in the conclusion highlights the educational benefits to use a system engineering method.

2. Overview of the method

This section presents the main steps of the methodology based on UML diagrams (Bock, 2009), (Weilkiens, 2008) that the students have to follow. In a "V" development cycle, engineering processes cover the usual activities of a top-down process: (1) definition of

stakeholders' needs, (2) definition of technical requirements, (3) design of functional and (4) physical architectures (see figure 1).
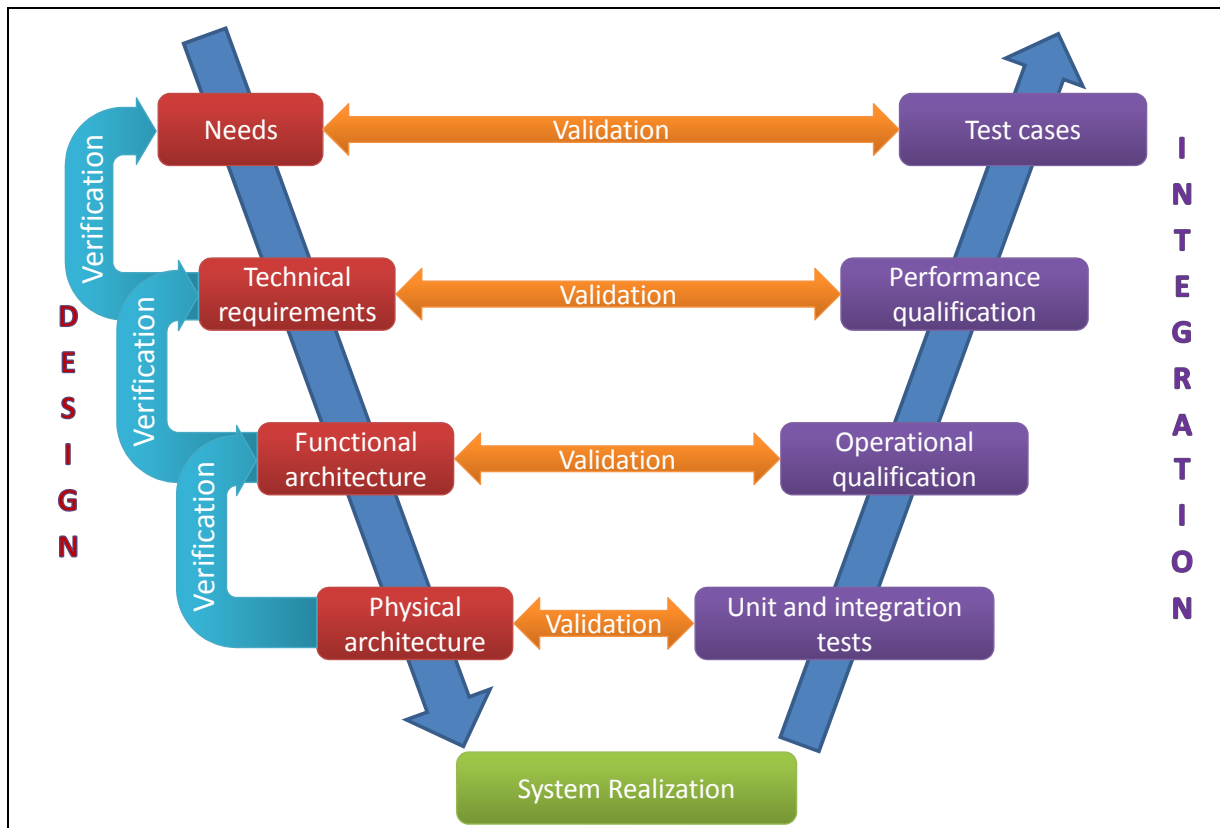

Fig. 1. Typical "V" cycle development

The *definition of stakeholders' needs* process first consists in identifying what kind of information the customer has given, generally a set of systems specifications which may not be either very well structured or even complete. The problem is then to understand the system in its specific context: define its purpose, mission, objectives and stakeholders, with the help of several operational scenarios. This process produces a document which lists and classifies stakeholders' needs and system constraints in every aspect of the system's use and lifecycle.

The goal of the *definition of technical requirements* process is to translate each need into an expression allowing the design of a feasible solution. It proceeds by refining the system mission and by breaking down the operational modes and scenarios into activities, in order to obtain corresponding technical requirements. This process also leads to complete and precise initial statements. The result is a document containing technical requirements that are coherent, formalized and verifiable (technical or physical features) and that will be useful for the designer.

After this essential step, it remains to build high-level *functional architectures*. The aim of this process is to establish and evaluate several functional architectures that could be candidates and retain one.

The *physical architectures* for the system, describing the chosen solution, as well as its physical interfaces, are given during the physical design processes.

At each step, a *verification process* is invoked, in order to justify the expression of needs, technical requirements, design choices, and to ensure traceability right through the development process.

Finally, a *validation process* is performed to compare technical requirements to performances obtained during in situ tests.

3. Description of the application

The students have to develop an application for the remote monitoring of several parameters of a room (luminosity, temperature) and of movements (intrusion detection). This application is based on nodes made up of a sensor, a microcontroller and a telecommunication module, in addition to the power module. Several rooms can be inter-connected while the sensors inside each room must interact, as illustrated in figure 2. Three categories of nodes are used according to the nature of the data they have to transmit. These data are different by their:

- nature: some are binary (detection of a threshold), others are analog,
- criticality,
- periodicity: some transmissions are periodic, while others are event-triggered.

As far as their transfer is concerned, these data have different needs concerning the quality of service. These needs must also be taken into account for the choice of a specific telecommunication technology and during the development of appropriated communication protocols.
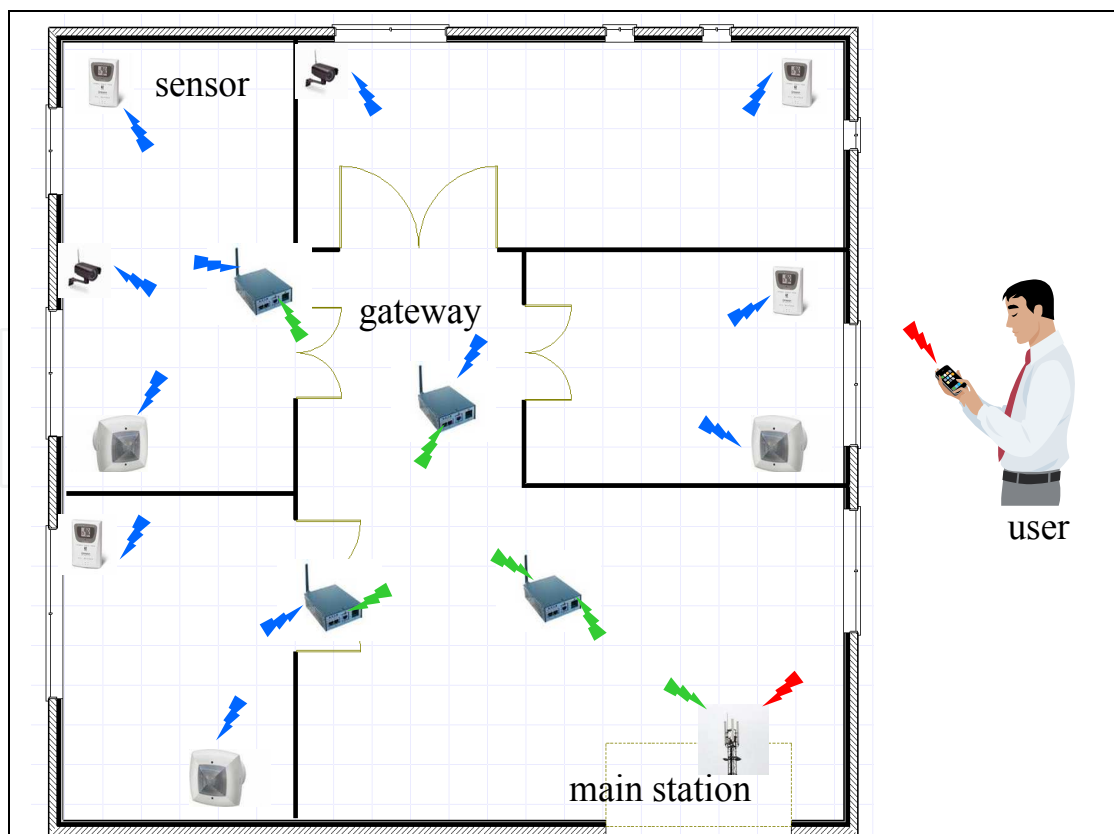

Fig. 2. Application of monitoring

4. Definition of stakeholders' needs and technical requirements

4.1 Needs

This step consists in enumerating the different elements of the context with which the system interacts when in use (physical and functional boundaries). The relationships between the system and these external elements are clearly illustrated in two distinct diagrams: a use case diagram, and an initial class diagram. The use case diagram is obtained by imagining global services showing the main interactions between the elements of the context and the system of interest. For example, in our surveillance application, the system collects energy readings from its environment and reports the collected temperatures to an operator; it is configured and repaired by a maintenance operator. On the basis of the use case diagram, we can draw up an initial class diagram containing the elements of the context and their physical links with the system of interest.

Students have to apply a system engineering (SE) method to design a sensor network. They use this network to validate their solution: choice of a telecommunication transceiver, communication and signalling protocols... suiting to a targeted application. This training includes 7 supervised sessions of practical works (3 hours each); free sessions are also scheduled so that the students can have access to the technical equipment. The starting point is the application specifications. Various items are available: software development tools, sensors (this teaching is synchronized with another one which objective is the development by the students of all the electronic part of the sensors), microcontroller evaluation boards and several kinds of transceiver with a detailed technical documentation. The interface boards between evaluation board and three transceivers are also given from the start.

At the end of this need identification process, they obtained two schemas with the main services provided or required by the system (figure 3) as well as the main components interacting with environment (figure 4).

4.2 Definition of technical requirements

Next step is to define what are the high-level stages of the system life and, in each one, what are the systems states (also called 'modes'). We usually find three cycles: upstream, utilization, downstream cycles. The upstream cycle includes four classical modes: design, realization, validation and installation. The utilisation cycle depends of the system of interest; for example, in our case, we distinguish maintenance, waking and monitoring states. In the downstream cycle, we usually find the retrieval mode.

Students are essentially involved in upstream and utilization modes. They obtain the general operational modes depicted in figure 5.

The technical requirements express the needs in the language of the project manager, or prime contractor, whereas the needs were previously expressed in the users' language. It is now necessary to complete and refine the information supplied by the users so that they lead to potential solutions. This is the goal of the technical requirements definition process.
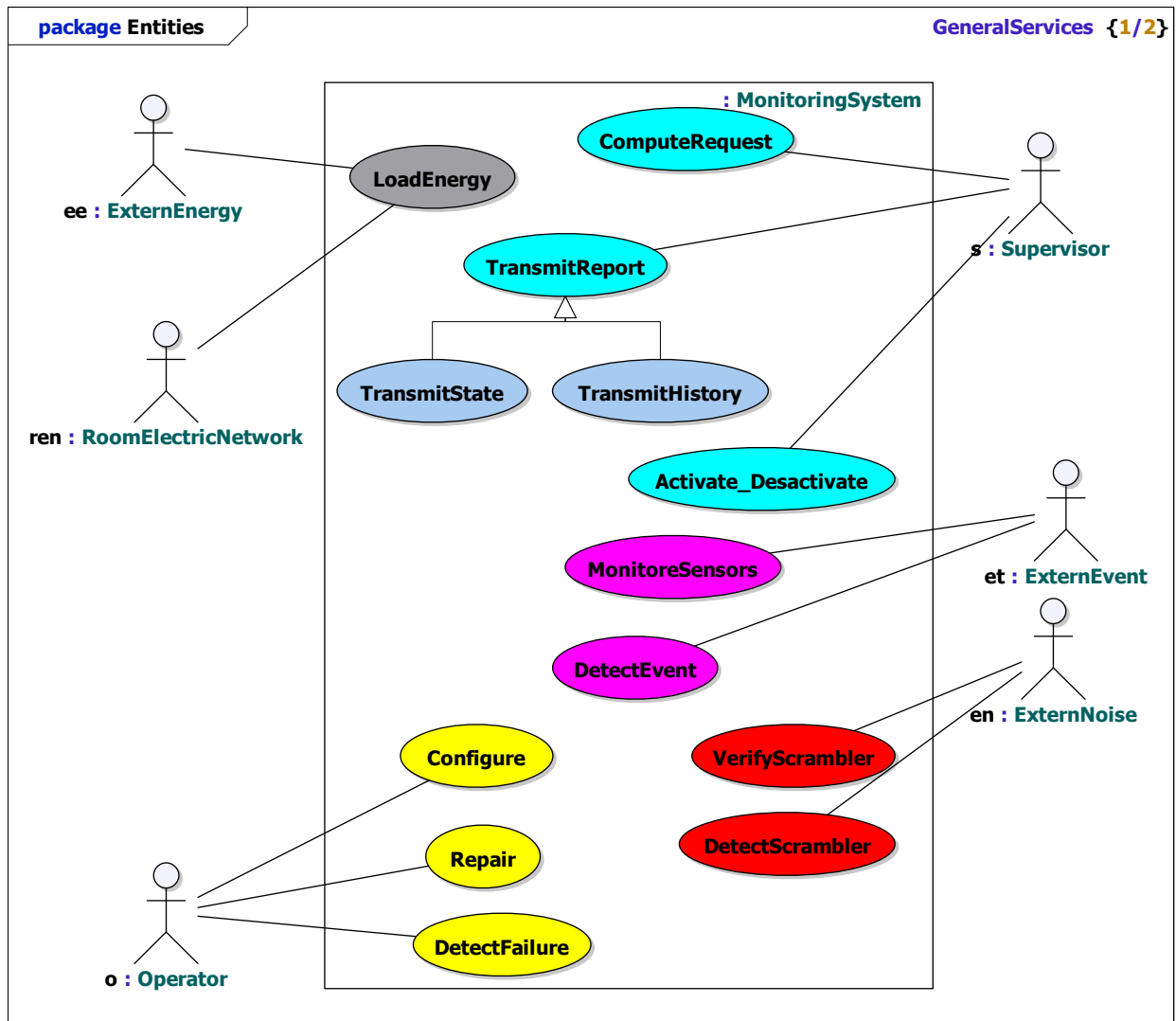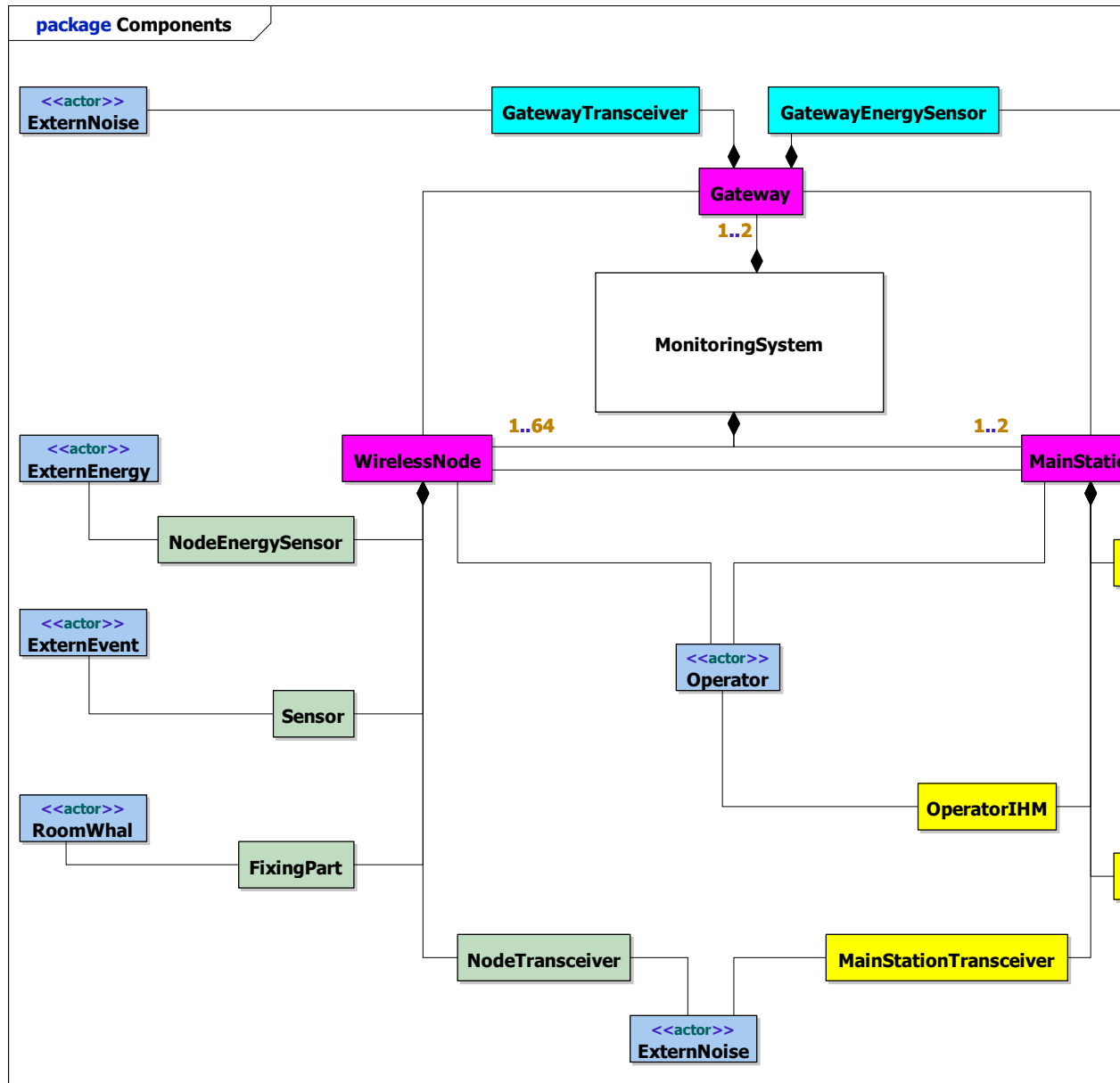
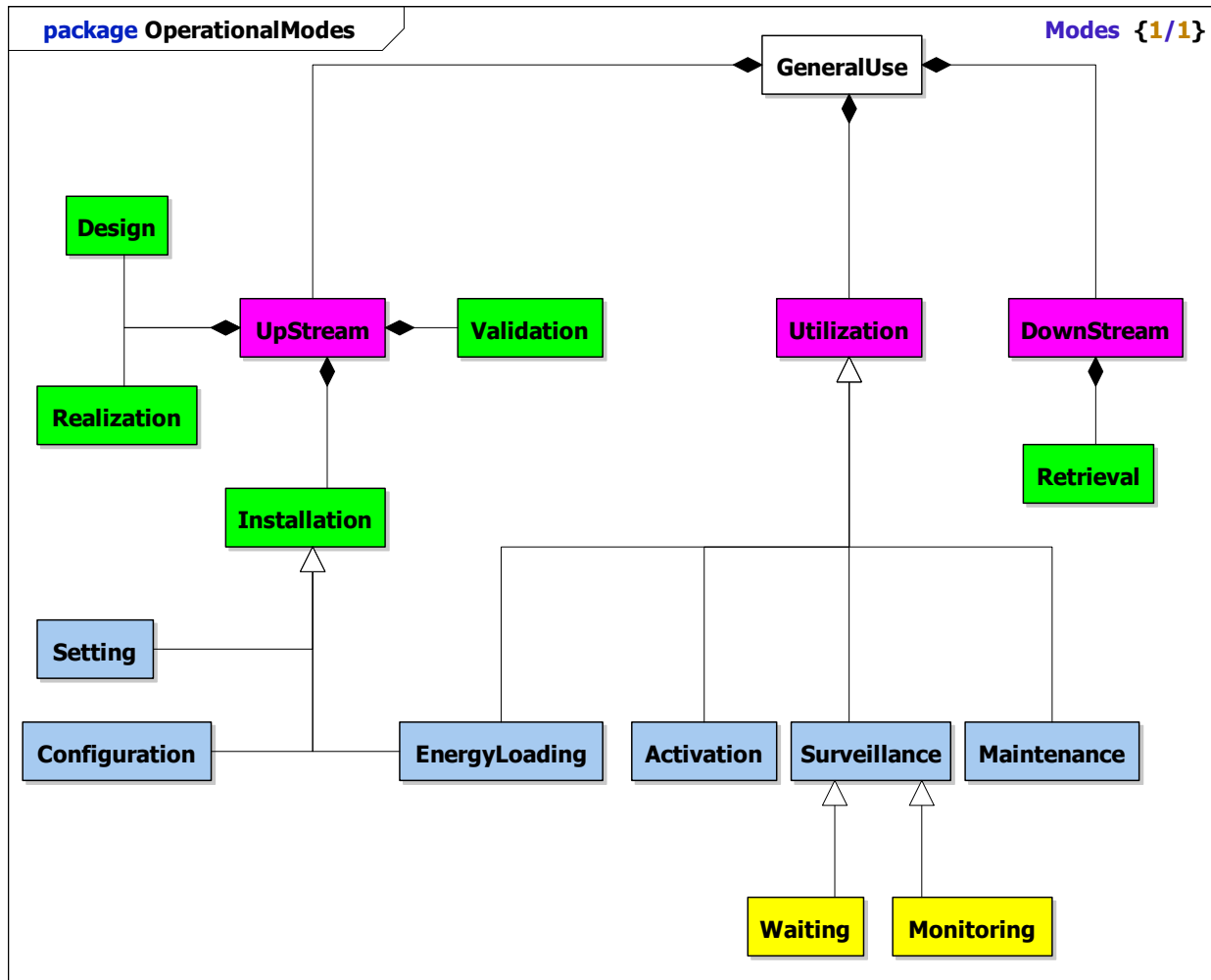Fig. 3. General service diagram

Fig. 4. Initial class diagram

Fig. 5. Operational modes

An example of technical requirements found by students and obtained by translating need into expressions allowing the design of a practicable/realizable solution is resumed in Table 1. Some previous works (Auriol *et al.*, 2008) explain a way to introduce students to requirement engineering.

Id	Needs	Technical Requirements
Operational Need 9	A node represents a position on the site	Define a unique ID for a node Define a process for associating a node with a position Define a specific grid for authentication Define a mechanism to upload node routing table

Table 1. Example of a need and definition of corresponding technical requirements

5. Design of functional architecture

The functional design process consists in identifying functional elements and designing functional architectures. The goal is to establish and evaluate several functional architectures that could be candidates and retain one. The identification of functional elements is directly obtained by an analysis of technical requirements: functional, interface and operational requirements, operational scenarios, and a breakdown of expected services. Performance requirements must then be allocated to functions. Once several functional architectures have been obtained, we need to identify and solve any conflicts between the elements of each functional solution (optimization process) and verify that each functional architecture correctly and fully satisfies the technical requirements. An evaluation of the various alternative functional architectures compared according to several parameters (quality, costs, times, performances, risks, etc.) leads to the best trade-off.

For example, when students deal with the services found in the first step, they obtain the Activity Diagram depicted in figure 6.

6. Design of physical architecture

Once a functional architecture for the system has been defined, the goal of the physical design process is to design various physical architectures to support these functions. The effort in this step is focused on identifying classes of components, establishing parameters and choosing criteria to assign the elements of the functional architecture to physical components, and the evaluation of several solutions. The physical architecture design process takes as its starting point the result of the functional design step, and refines it. Indeed, for each architecture, the first task is to decide whether the functional breakdown is sufficient to identify physical components and/or technologies capable of supporting the execution of the end functions of the functional architecture. The objective is then to consider various possible physical architectures and to estimate their feasibility. Once various possible physical architectures have been obtained, it only remains to choose a final architecture. Once this choice has been made, the final task is to fully specify the solution, to validate and justify it.

Students extract the components of the system from the initial class diagram (figure 7) and progressively complete the physical architecture diagram with components according to the functions found during the precedent step (figures 8&9).

At this level of breakdown, students add the available solutions. In this chapter, we only give some details about transceivers and communication protocols. For example, they can choose transceivers among:

- a half-duplex FM transmitter, manufactured by Telital, using FSK modulation at 433MHz
- a EEE 802.15.4 [9] transmitter with a *ZigBee* [10] stack manufactured by Microchip
- a GSM / GPRS modem manufactured by Sagem

The choice of these technologies is driven by the diversity of the services that they offer. To simplify, four technical parameters are retained: cost, range, consumption and access BUS and the embedded Medium Access Control (MAC) layer. During the integration step, students have to refine and to extend these performances
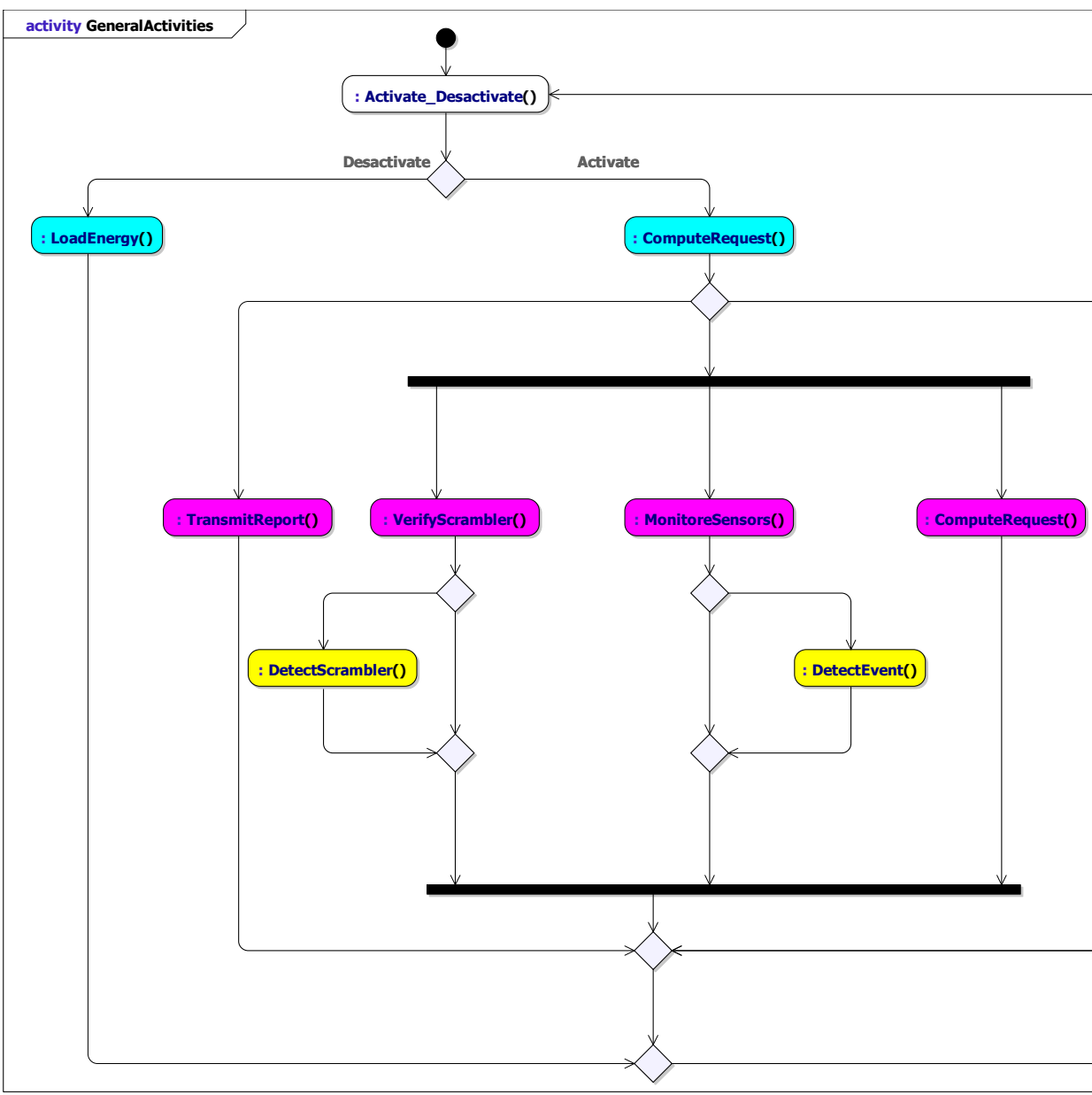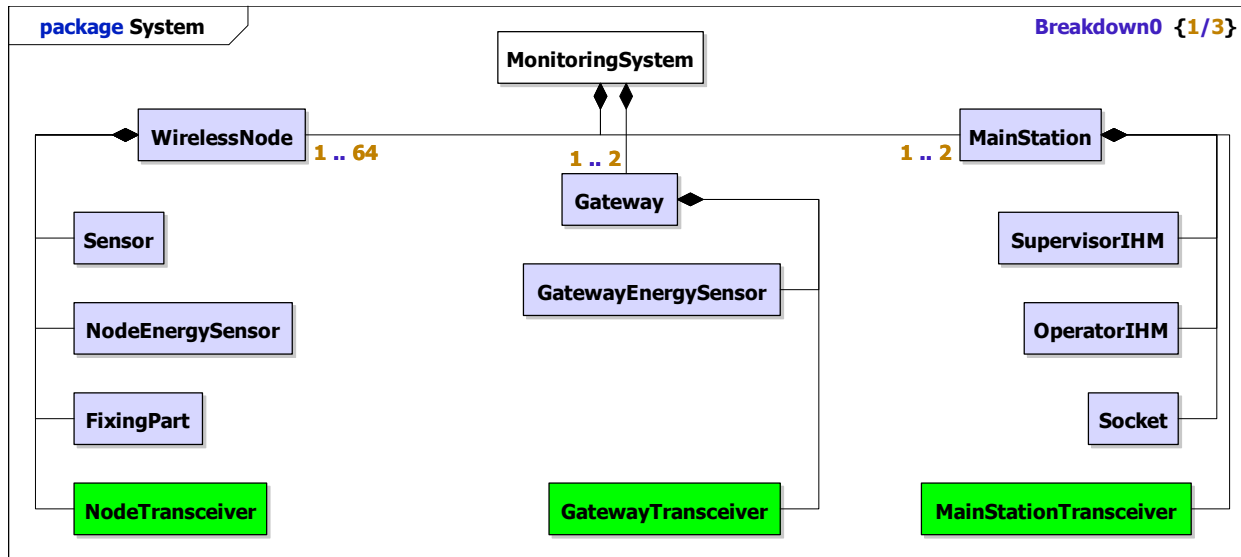
Fig. 6. General activity diagram
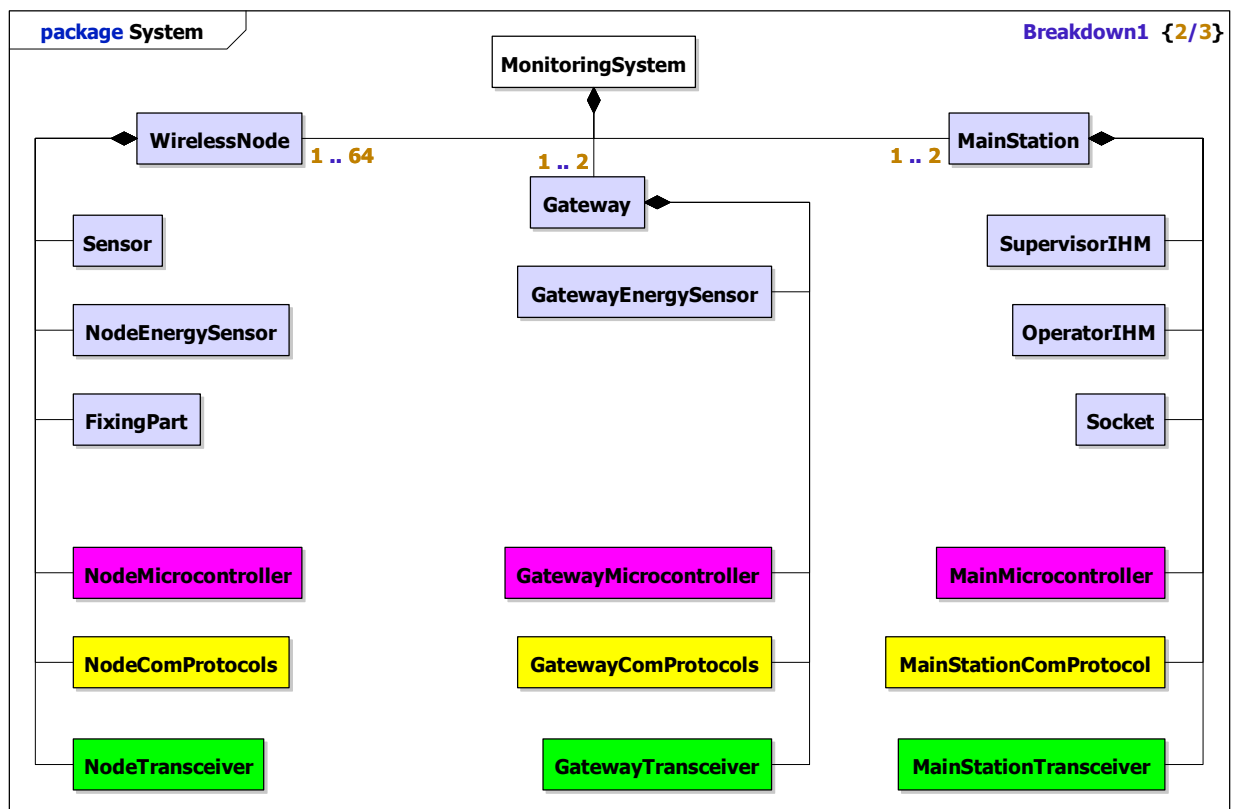
Fig. 7. Main components

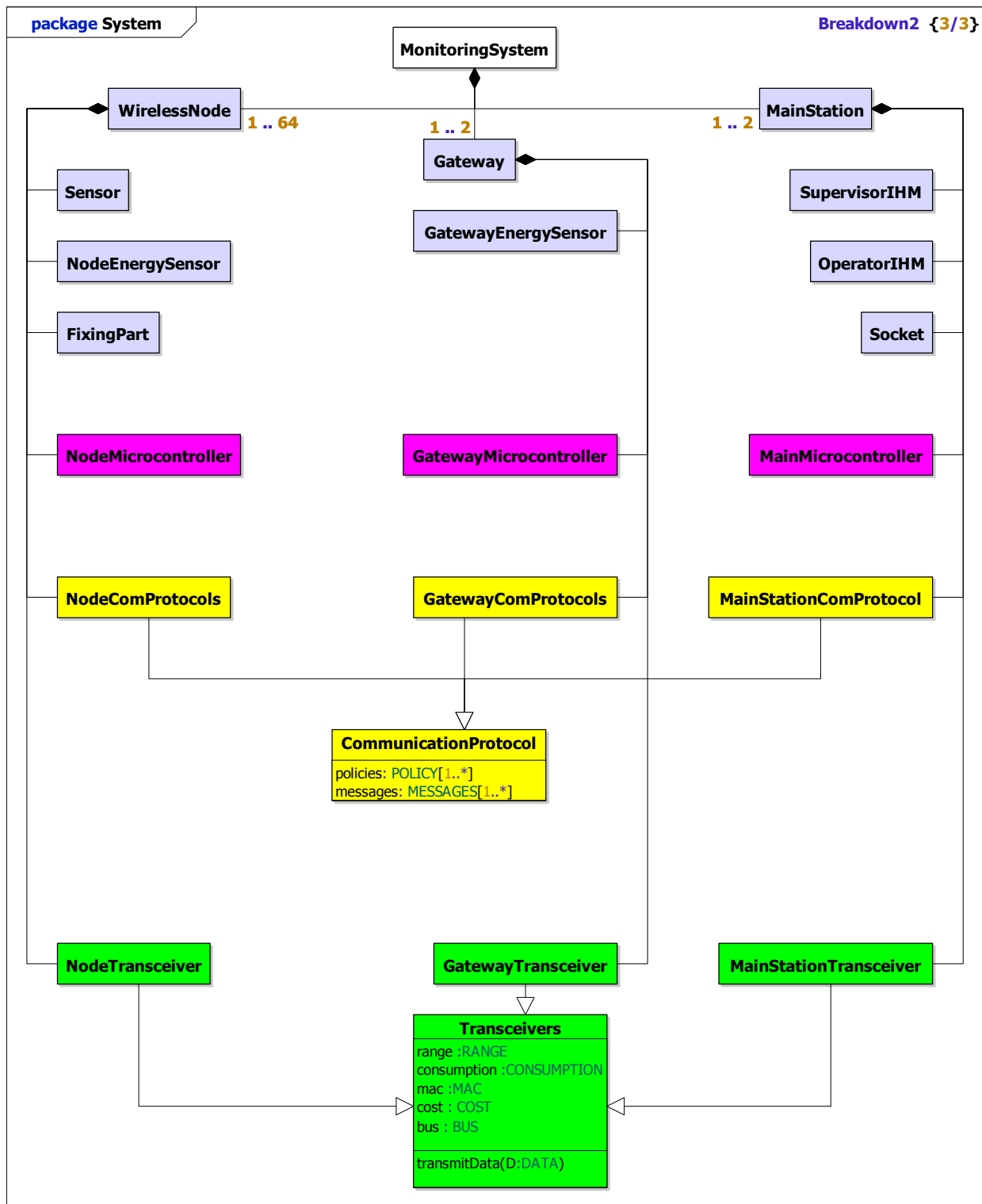

Fig. 8. Microcontrollers and protocols components

Fig. 9. Final class diagram

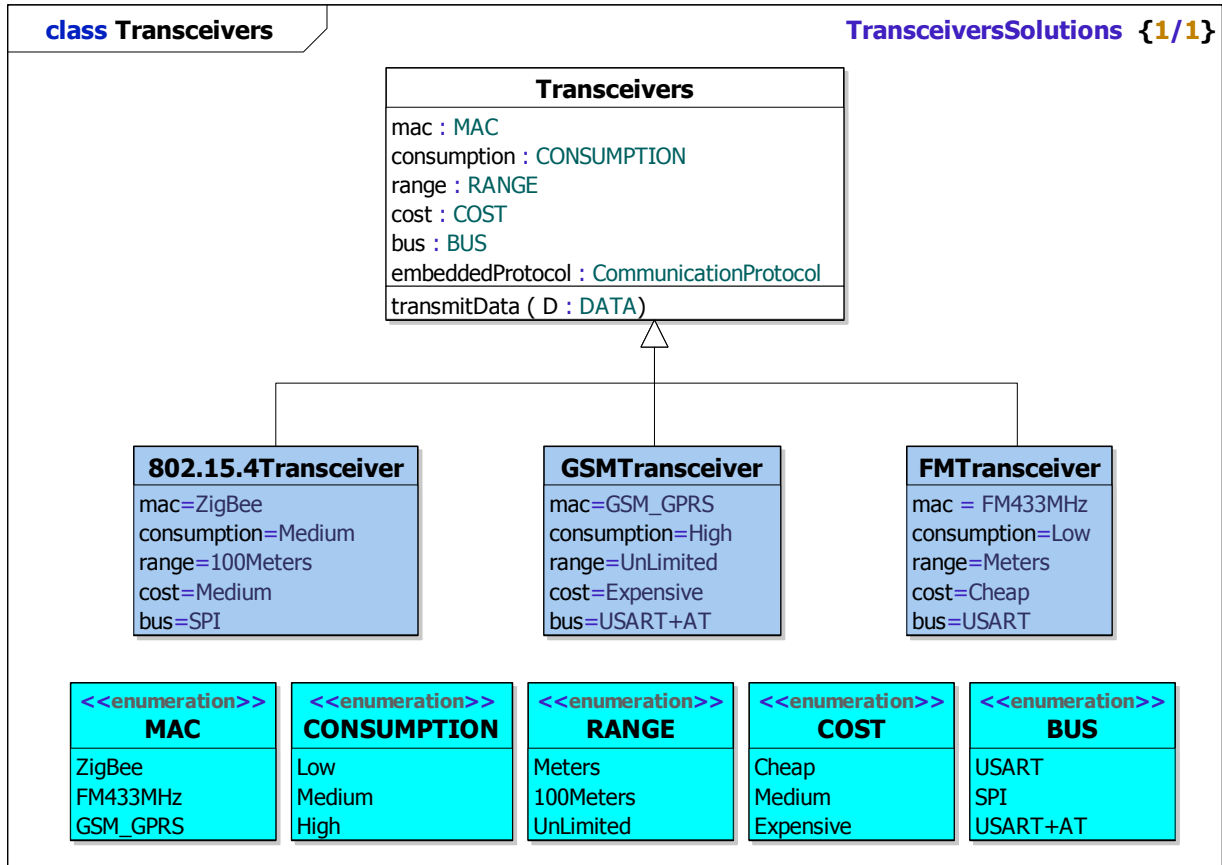Then, students obtain the schema in figure 10.

Fig. 10. Details of transceiver diagram

Each transceiver studied has specific data transfer and signaling protocol which can be extremely basic, or may include complex embedded applications. The implementation of these services also differs a lot from one device to another. To develop the application of remote monitoring, students must initially understand the need for these services, and then extend them if necessary. Mainly, the retained parameters to characterize the communication protocols concern:

- services as: connection, loss detection, carrier detection, acknowledges,
- policies of deployment,
- reliability to transmit several kinds of messages which could be analog or binary, critic or not, periodic or not.
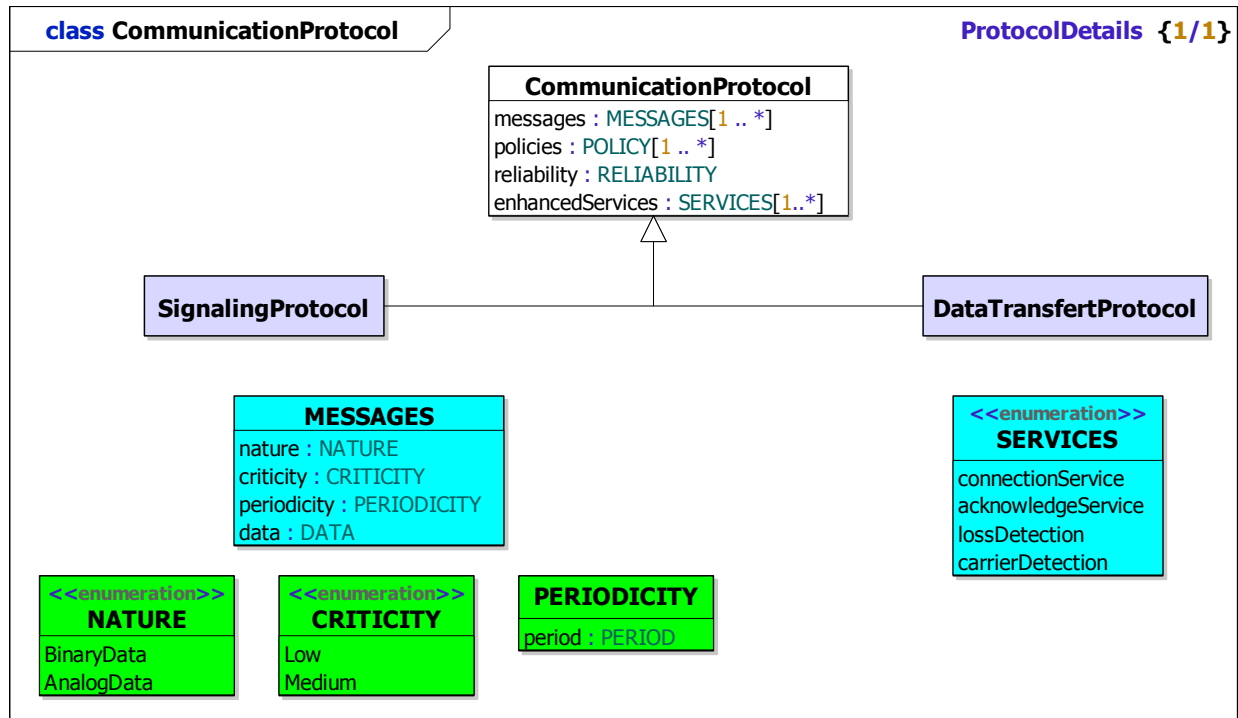
Students obtain the schema of figure 11.

Fig. 11. Details of protocol diagram

7. Integration and validation

During this step of integration, students connect component via customized boards to the evaluation board of a microcontroller (MCBSTM32 [7] of Keil, processor ARM / ST). This configuration represents a "less embedded" solution than a dedicated electronic board which would have specifically been developed and offers a high flexibility of use and of debug to the students by supplying a multi line LCD screen for display, free zones for oscilloscope measures, series connections for connections to a terminal PC, diodes... The data transfer and signalling protocols are implemented on the microcontroller by means of the environment µVision3 of Keil [8] which offers functions of simulation and transfer towards a microcontroller.

Students have to discover and understand the manipulation of each device taken separately before starting to completely implement the platform to validate their choice of components.

The students follow the evolution detailed below:

Step 1. discovery of devices. For that purpose, they test every device by establishing a direct and basic communication (send of an *ASCII* characters) between a single transmitter and a receiver. This step is common to every device and requires no elaborated configuration nor to add services to those already offered by modules.

Step 2. several transmitters on the network. By testing services offered by the first device (FM technology), students understand that it is necessary to add a field "address"

in any emission of information. They also note a risk of collision and loss of frames which grows with the transmitter number if they do not use a device offering suitable services or if they do not extend those basic ones proposed by the device.

Step 3. the network spreads out. The initial range of the first two modules (FM and *ZigBee*) is not sufficient to cover communication needs on more important distances. It is then essential to set up several modules with a suitable signaling service on intermediate devices.

Gradually, the students thus take into account a more and more complex topology until they consider the complete platform of test compatible with the application of remote monitoring. The objective for the students is to validate their choice of components by the mean of a platform whose technological solution is represented in figure 12. Indeed, this solution appears as the best compromise if all the parameters are taken into account.
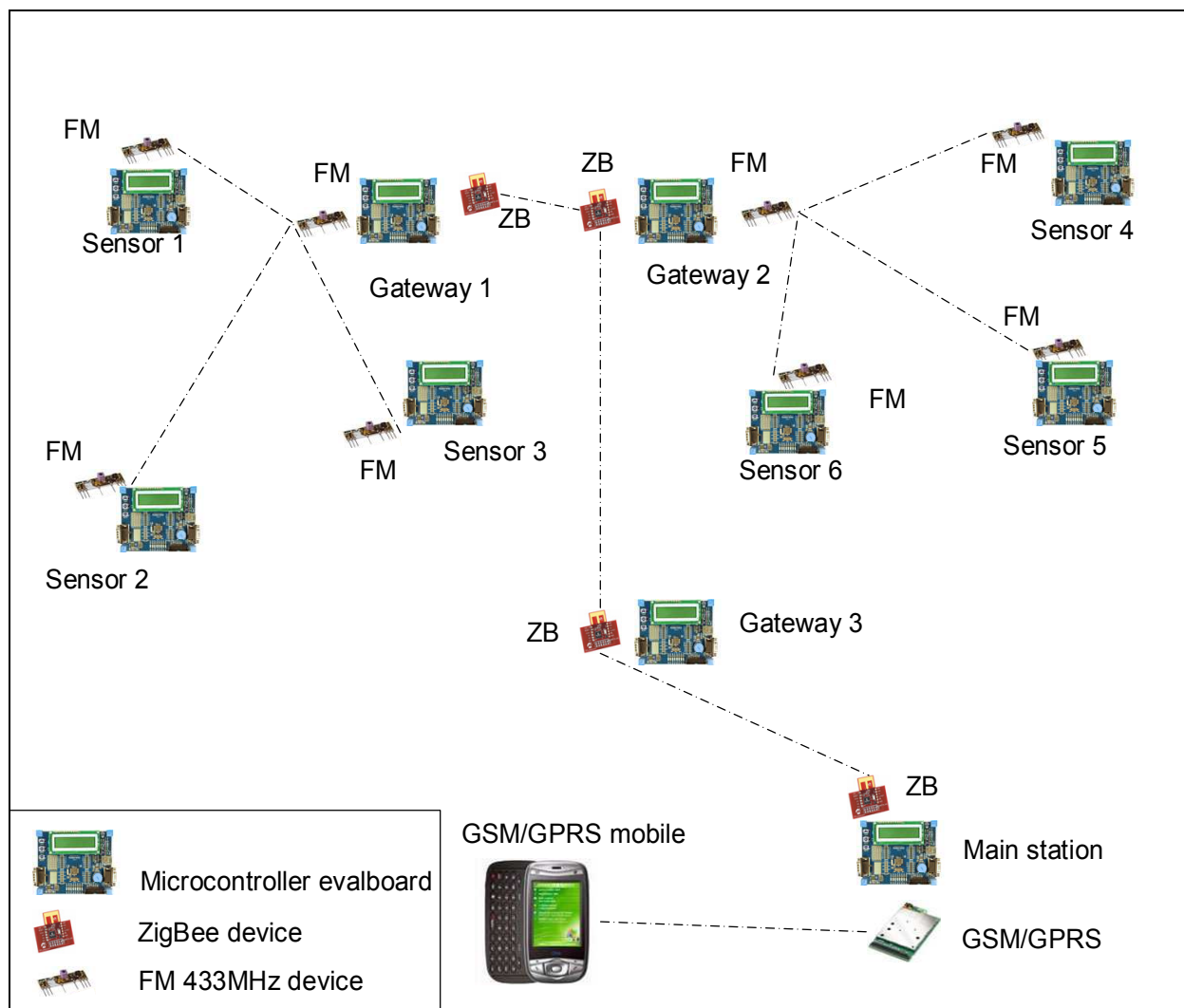

Fig. 12. Technological solution

8. Conclusion

This chapter describes a teaching experiment during which the students apply a systems engineering approach to design a solution for a complex system when numerous design and implementation options are available. The chosen application is the remote surveillance of several rooms simultaneously taking into account three parameters: light, temperature and movement. This application is based on wireless terminal nodes composed of a sensor, a microcontroller and a telecommunication module. They dispose of a set of off-the-shelf software and hardware components from which they must design the best functional and architectural solutions, by drafting a technical requirements dossier to satisfy the users' needs.

For that, they are guided to progress following the steps of the V cycle. They start by studying the requirements to extract an exhaustive list of needs. Then they propose and evaluate functional and architectural solutions. They finally implement the chosen solution by integrating the different modules of the physical architecture in order to validate their 'systems engineering' approach.

This experiment was positive in that it taught students that even if they had no previous specific knowledge of the field of wireless Personal Area Networks, a formalised systems engineering approach allowed them to develop a solution.

9. Acknowledgement

10. References

ANSI/IEEE 1220, 2005, Standard for Application and Management of the Systems Engineering Process, www.ansi.org

Auriol G., Baron C., Fourniols J-Y., 2008, Teaching requirements skills within the context of a physical engineering project, *The Third International Workshop on Requirements Engineering Education and Training (REET'08)*, September 9th, 2008, Barcelona, Spain

Bock C., 2006, SysML and UML Support for Activity Modeling, *Systems Engineering*, Vol. 9, Number 2, Summer 2006.

Brownsell S. J., Williams G., Bradley D. A., Bragg R., Catlin P., Carlier J, 1999, Future systems for remote health care, *Jour. Telemedicine & Telecare*, vol 5, pp 141-152

Honour E. C. , 2004. Understanding the value of systems engineering, *INCOSE International Symposium*, Toulouse, France.

ISO-IEC 15288, 2008, System Life Cycle Processes, www.iso.org

Kotz1 D. and Essien K., 2005, Analysis of a Campus-Wide Wireless Network, *Wireless Networks*, Vol. 11, Numbers 1-2, pp 115-133.

Martin J. N., 1998, Overview of the EIA 632 Standard – Processes for Engineering a System, *17th Digital Avionics Systems Conference (DASC)*, 31 Oct-7 Nov 1998, Bellevue, WA, USA

Terry A. Bahill and Clarck Briggs. "The systems engineering started in the middle process :
        A consensus of systems engineers and project managers", Systems Engineering,
        4(2) pages 156–167, 2001.

Ullrich K.T. and Eppinger S.D., "Product design and development", McGraw Hill
        International Edition, 2003

Weilkiens T., 2008, *Systems Engineering with SysML/UML: Modeling, Analysis, Design*, Morgan
        Kaufmann Publishers In, 28 March 2008, The MK/OMG Press, ISBN 0123742749,
        320p

Systems Engineering - Practice and Theory

Edited by Prof. Boris Cogan

The book "Systems Engineering: Practice and Theory" is a collection of articles written by developers and researches from all around the globe. Mostly they present methodologies for separate Systems Engineering processes; others consider issues of adjacent knowledge areas and sub-areas that significantly contribute to systems development, operation, and maintenance. Case studies include aircraft, spacecrafts, and space systems development, post-analysis of data collected during operation of large systems etc. Important issues related to "bottlenecks" of Systems Engineering, such as complexity, reliability, and safety of different kinds of systems, creation, operation and maintenance of services, system-human communication, and management tasks done during system projects are addressed in the collection. This book is for people who are interested in the modern state of the Systems Engineering knowledge area and for systems engineers involved in different activities of the area. Some articles may be a valuable source for university lecturers and students; most of case studies can be directly used in Systems Engineering courses as illustrative materials.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Guillaume Auriol, Claude Baron, Vikas Shukla and Jean-Yves Fourniols (2012). System Engineering Method for System Design, Systems Engineering - Practice and Theory, Prof. Boris Cogan (Ed.), ISBN: 978-953-51-0322-6, InTech, Available from: http://www.intechopen.com/books/systems-engineering-practice-and-theory/system-engineering-method-for-system-design

INTECH
open science | open minds