

SysML based Design for Variability enabling the Reusability of Legacy Systems towards the support of Diverse Standard Compliant Implementations or Standard Updates: The Case of IEEE-802.15.6 Standard for e-Health Applications

Sofia Meacham

Bournemouth University
Fern Barrow, Poole, Dorset, BH125BB
+44(0)1202965115

smeacham@bournemouth.ac.uk

Fotios Gioulekas

University General Hospital of Larissa
Mezourio, Larissa, GR 41110
+30-2413-501450

gioulekas@uhl.gr

Keith Phalp

Bournemouth University
Fern Barrow, Poole, Dorset, BH125BB
+44(0)1202965571

kphalp@bournemouth.ac.uk

ABSTRACT

The aim of this paper is to provide a consistent development path enabling the re-usability of in house legacy systems or architectures towards their re-design, in order to ensure compliance with evolving standards, by using the new features of SysML for modelling variants. Modern standards evolve quickly, include advanced functionalities and operations and support diverse implementations. System industries need to cope with such standards changes by modifying their current technologies. This paper shows how a novel engineering process (SysML modelling) could be employed to define consistently the specification and the migration procedure of legacy systems to their variants. Within this work SysML characteristics such as package and block diagrams, are employed, with an emphasis on variability modelling, as a basis for standard compliant architecture implementation, thus providing design flexibility and reusability at several abstraction levels. As an illustration of our proposed method we present models of two variant Physical Layer structures for IEEE-802.15.6 Standard for e-Health Applications. The advanced SysML features are used to target the re-usability of a legacy Narrow-Band (NB) physical layer subsystem for the Wireless Body Area Network standard and to implement the alternative Ultra-Wide Band (UWB). Therefore, we contend that such methods bring potential benefits to those needing to ensure compliance when producing product variants.

Categories and Subject Descriptors

D.2.13 [SOFTWARE ENGINEERING]: Reusable software – reuse models. C [Computer Systems Organization]: General –

Systems specification methodology.

General Terms

Design, Standardization, Languages.

Keywords

Variability, standard compliance, standard evolution, SysML, Eclipse\Papyrus, eHealth.

1. INTRODUCTION

Evolving technology and market trends lead international organizations to modify, extend or update current standards in order to cope with changing needs. It is also common that several standards like IEEE P1901 [1] and IEEE 802.15.6 [2] propose alternative approaches for Physical Layer implementations. The latter characteristic forces system industries to support these diverse and variant functionalities, to maintain or increase their market share. The majority of the system design houses already have a legacy system or subsystem in their portfolio that is used as basis towards their effort to comply with the standard diverse specifications and design the next variant subsystem. However, during the design product cycle, there is a gap in the methodological step that could lead the total engineering effort towards the re-use of the in house available system, delineate its basic functionality, modify the current features according to standard alternative functionalities and provide a consistent and abstract formal model which is opted to serve as specification. Indeed, it was the authors' own experiences in international telecommunication industries, such as INTRACOM S.A., SGS-Thomson Microelectronics, that motivated us to attempt to find a solution to these conflicting business needs. Hence, the objective of the proposed work here is to exploit SysML as an engineering tool to define consistently the specification and the migration procedure of legacy systems to their variants. By the use of SysML, and its new features, to support model-driven design with variability, we hope to close the aforementioned error-prone gap [3]. To this end, the physical layer variant of the IEEE 802.15.6 standard for e-health applications is chosen as a case-study [4]. Distinctively, it is shown how a legacy NB physical layer

subsystem is re-used to develop the UWB alternative conformed to the standard specifications.

Research work on using UML for Wireless Body Area networks has been done previously [5]. However, for our work we are proposing SysML as a modelling language. SysML is selected due to its suitability for modelling systems and model-based design [6]. It was the result of a UML RFP recommendation for system engineers and has been adopted by PMG since 2006. It offers system engineers several noteworthy improvements over UML. SysML reduces the software-centric restrictions of UML and adds more diagram types such as block definition diagrams, internal block diagrams, parametric and requirements diagrams. Due to the above additions, SysML is able to model a wide range of systems such as hardware, software, information, processes [7].

In section 2, an overview of the importance of standard compliance and standard evolution is given along with its impact on existing legacy systems. Section 3 argues the need for model-based design and variability modelling to enable reusability of legacy systems. In section 4, current approaches to variability modelling are presented, which lead to section 5 where our proposed approach for variability is introduced. We propose a method and corresponding metamodel based on SysML modelling. In section 6 and 7, our case study is described along with the resulting SysML descriptions. In section 8, conclusions and future work are presented.

2. STANDARDS COMPLIANCE, STANDARDS EVOLUTION AND LEGACY SYSTEMS

Standards compliance is a very important and necessary aspect for a range of industries, and particularly so for our exemplar of companies that develop Wireless Area Networks. It is through the use of standards that the requirements of interconnectivity and interoperability can be assured [8].

However, with systems evolution and changes of requirements, standards are also evolving [9]. Systems houses need to be able to adapt their designs to new requirements in very short timescales. Industries are faced with the problem of how to conform to new standard requirements without losing previous designs.

Therefore, a method to address evolving standards compliance by reusing as much as possible from the previous designs or implementations (legacy systems), would be an extremely attractive goal for industrial competitiveness and viability.

3. RATIONALE FOR MODEL-BASED DESIGN AND VARIABILITY MODELLING

3.1 Model-based Systems Engineering

Model-based Systems Engineering (MBSE) is an approach to the design and development of systems in which models take a central role, not only for analysis of these systems but also for their constructions [10]. MBSE is part of a long-term trend towards model-centric approaches adopted by other engineering disciplines including mechanical, electrical and software engineering [11] [12].

According to INCOSE, the adoption of MBSE has several advantages [13] such as: improved communication between stakeholders, team members through diagrammatic model representations; improved quality through early identification of problems and fewer errors at the integration stage; increased

productivity through reusability of existing models and reduced risk through improved estimates and on-going requirements validation and verification. Overall, it has been said to increase productivity and efficiency in the design and development mainly of complex systems.

3.2 Variability as Product Line Engineering (PLE) approach

On the other hand, reusability is a very well-known concept and several attempts to design/develop with reusability in mind have been developed over the years such as IP reuse, code reuse.

Among them and originally developed by the automotive industry is “*product line engineering*”. Most automotive software manufacturers supply not just one but many customers. Albeit customers may use the same type of Electronic Control Unit (ECU), their controller software is likely to vary due to different customer requirements [14]. For instance, gentle gear shifting is preferred for one type of cars, but more sporty shifting for another. Software product lines consider related products, their commonalities and variability. Variability makes the main difference when comparing product lines with single systems [15].

So concluding, in this case study, our problem is implementing reusability for systems that conform to evolving standards. The authors of this paper believe that the basic ideas behind model-based design and product line engineering would be applicable and beneficial for developing evolving standards compliant systems. In both cases, there is a core functionality identified and variation points for the different parts of the system and different routes have to be followed according to the specific product line or standard to which the variant must conform. Therefore, in this paper the use of variability modelling for conformance to evolving standards is proposed.

4. VARIABILITY MODELLING APPROACH

There are several techniques for modelling variants that are most commonly used such as decision tables, decision trees, and feature diagrams [16]. Decision tables and decision trees are easy to understand and to use as they don't require a new modelling language. However, they become large and unclear when there are numerous variants. Feature diagrams also use a separate notation and after familiarization they are easy to understand and to use, and formal proofs are enabled through the existence of specialized tools such as pure::variants. Note that it is important to distinguish between features and variants. While features are characteristics of a system (at a high level descriptions), variants occur only when two or more systems have different features implemented (at the implementation level).

From the above, we can conclude that the majority of existing notations for modelling variants have the major disadvantage of using a different notation for the variants than the actual system description and in most cases a formal proof is not possible.

In the authors' opinion, a new approach to modelling variants requires a different approach. Integration between the variant and the system models instead of separating them will provide a unified environment for modelling systems with variability. This will enable the sustainability and maintainability of the system as any changes in the variants or in the system belong to the same model. On the other hand, if we had separate models, an enormous amount of additional effort would be required in order

to synchronize them and maintain consistency during the product lifecycle. Hence, we propose to model variants of the system together with its requirements, its structure, its functionality, its realization or its service packages. For this, we use a modelling language to model our system, namely SysML, and extend it with necessary stereotypes for modelling variants.

To conclude, modelling variants for standard-compliant systems requires a new approach, able to cover the entire engineering process. SysML has been chosen because it provides a comprehensive support for modelling, covering all the required phases.

5. PROPOSED SYSML METAMODEL FOR VARIABILITY

In this paper, we propose to use SysML modelling language without any major extensions and/or modifications [17]. We suggest a particular method for using existing SysML constructs in order to describe variability and provide their corresponding meta-models. This method has the advantage that the designer will not have to learn a new notation in order to describe variability. The variability is depicted in the same model as the system model which has been proven to be beneficial as opposed to separate models.

Alternatively, we could extend SysML to describe variability by creating a new SysML profile. That would mean that we would have to introduce and propose new language constructs. This approach would be beneficial if the language changes and extensions were major. This is not the case for our application as, according to the proposed method, only a small number of SysML constructs are adequate to model variability for our application. It is a subset of the method proposed by [16], tailored to the requirements of our application of design or develop evolving standards compliant systems.

The proposed method consists of modelling variability at three levels: the package level where each package consists of several diagrams such as block and requirements; the block level where we are modelling system blocks; and the internal block diagram level, where we model internal blocks and dataflows (signals) between them. Note that for each of these levels, our description can be done in a hierarchical way as was also described in [18]. These levels of variability modelling will be described in the following sections.

5.1 Package level variability

First, we consider modelling variability at the package level. Packages are modelling constructs that can contain other SysML constructs such as blocks and requirement diagrams. Using packages for modelling enables reusability of whole parts of the design and it is recommended for the design of large and complicated systems. The method for package variability is presented in Figure 1:

Three main options are encountered when we model variability in our application:

Case A (Mandatory):

In this case, there is some core functionality that will definitely be included in a variant. For this purpose, we suggest the use of `<<import>>` SysML construct for packages.

Case B (Optional):

In this case, the core functionality might be included or not to a set of variants and it is a common optional functionality. For this purpose, we use SysML inheritance construct where the variant packages “inherit” the common functionality. According to the

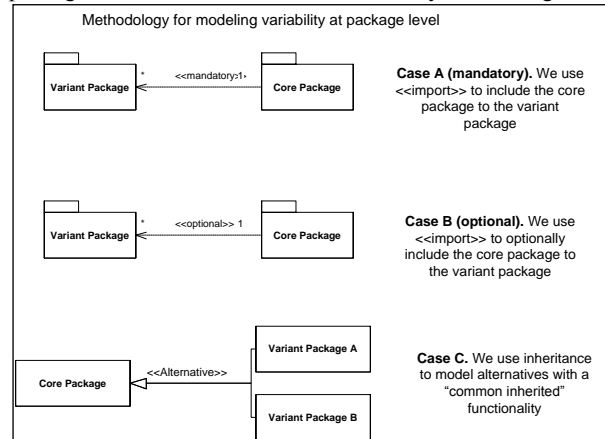


Figure 1. Methodology for package variability

SysML inheritance definition, this means that variant might use or not that common core functionality.

Case C (Alternatives):

In this case, the core functionality might be inherited in several different variants that can be considered as alternatives.

Note that for the above we use existing SysML constructs with the difference that we have to define two new stereotypes: mandatory and optional for the `<<import>>` relationship.

The proposed metamodel for package variability is presented in Figure 2:

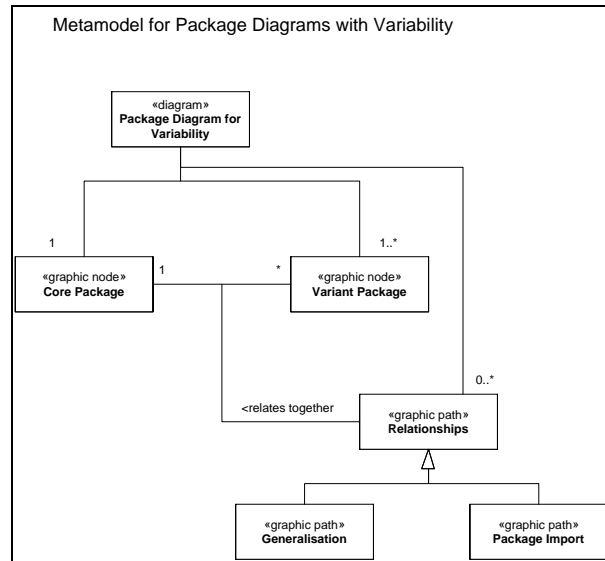


Figure 2. Metamodel for package variability

In this metamodel, the two ways of modelling package variability in SysML are depicted. For Case A-mandatory, we propose the use of package import which is standard SysML type of relationship between packages [19]. For Case B – optional, we propose the SysML generalization type of relationship for modelling the “possibility” of including some core functionality to variants.

5.2 Block level variability

The methodology for block variability is presented in Figure 3:

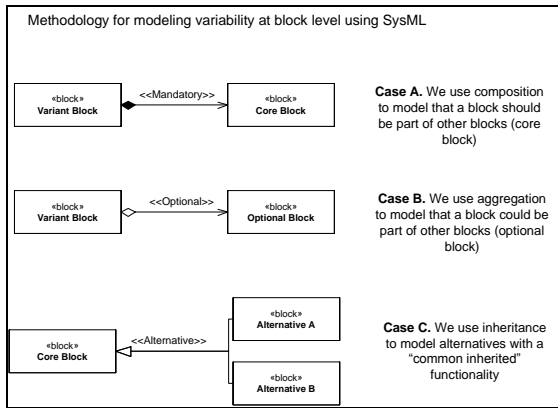


Figure 3. Metamodel for block variability

Case A (Mandatory):

In this case, we use *SysML composition* to show that a core block should be part of other variant blocks. According to [20], the core block that is the target of the composition relationship cannot exist in its own right and has to be part of the variant block. This consequently implements the “mandatory” requirement for a variant-core block relationship.

Case B (Optional):

In this case, we use *SysML aggregation* to show that an optional core block might be contained in a variant block. Aggregation is used for this case as composition cannot be optional. With aggregation the end part of the relationship (optional block) can exist in its own right and doesn’t have to be part of the variant block [20].

Case C (Alternative):

In this case, we use *SysML generalization* to model alternative variants. For example, the core functionality could be inherited in several different versions/variants of the system which could be alternative implementations in order to model different options in a standard. The metamodel for block variability is presented in Figure 4:

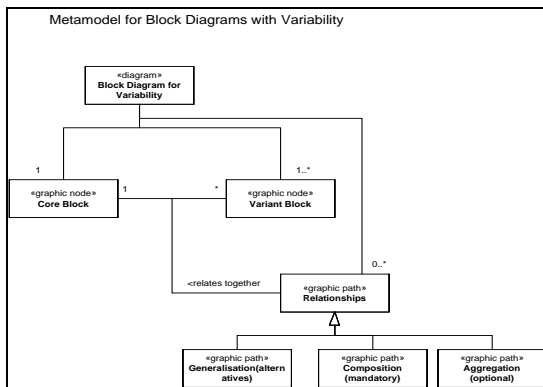


Figure 4. Metamodel for block variability

In this metamodel, the three ways of modelling block variability in SysML are depicted. For Case A-mandatory, we propose the use of composition to model that it is mandatory that a variant block is composed by the specific core block. For Case B –

optional, we propose the SysML aggregation type of relationship for modelling the “possibility” of including some core functionality to the variant block. For Case C - alternatives, generalization is used to model that several alternatives might be inheriting the core functionality.

To summarise, in this paper we are proposing to model variability using existing and simple SysML constructs such as composition, aggregation and generalization. The case study illustrates that if these constructs are used according to the previously defined metamodels, we can effectively model variability for systems whilst maintaining compliance to evolving standards.

6. CASE STUDY: THE CASE OF IEEE-802.15.6 STANDARD FOR E-HEALTH APPLICATIONS

A common set of healthcare technologies are based on the utilization of wireless sensors for remote monitoring of patient care without restrictions on body normal activities. These sensors, which are attached to/on/in the human body and measure vital signals e.g. heart beats, blood pressure, glucose levels etc., transmit their data using wireless physical structures forming a wireless body area network (WBAN) [21]. The demand for continuous monitoring of patients using WBAN for short-range wireless communications in the vicinity of, or inside, the human body with certain criteria of low-power, quality of service (QoS), security, and reliability [22] has prompted standardization bodies to proceed with an appropriate standard model. The IEEE 802 group published in 2012 the IEEE 802.15.6 communications standard [2]. It standardizes the wireless communication in the vicinity of, or inside, a human body (but not limited to humans) exploiting existing industrial scientific medical (ISM) bands as well as frequency bands approved by national medical and/or regulatory authorities. It exercises networks with wearable computing devices that support QoS, extremely low power, and data rates up to 10 Mbps is required while simultaneously complying with strict non-interference guidelines where needed. This standard considers effects on portable antennas due to the presence of a person (varying with male, female, skinny, heavy, etc.), radiation pattern shaping to minimize specific absorption rate (SAR) into the body, and changes in characteristics as a result of the user motions.

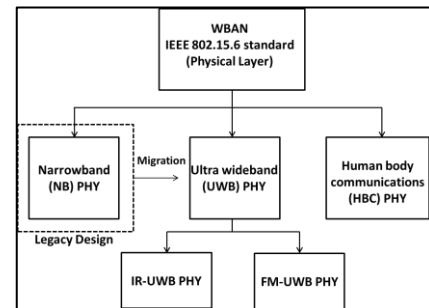


Figure 5. IEEE 802.15.6 Physical Layer classification

The IEEE 802.15.6 standard defines a Medium Access Control (MAC) layer that supports three diverged Physical (PHY) layers, i.e. Narrowband (NB), Ultra-wideband (UWB), and Human Body Communications (HBC) layers. Figure 5 depicts the PHY layers that IEEE 802.15.6 standard specifies. The NB PHY utilizes low control overhead, very low peak power consumption and robustness against interference based on DPSK modulation formats, BCH coding and suitable pulse shaping while (UWB)

PHY is classified in two categories (i) IR-UWB as mandatory PHY and (ii) FM-UWB as optional PHY depending on the modulation order and scheme. UWB PHY transmits information over a large bandwidth. On the other hand the HBC PHY uses the electric field communication technology and exploits the human body as a means of propagation for the data transmission, through the galvanic coupling of signal currents.

In this case study, it is considered that a legacy NB PHY system has been developed by a design group [23] [24], and it is requested that this design is migrated to the UWB PHY, as illustrated in Figure 5. In [25], it has been shown that RF SysML modelling enhances productivity because it captures the requirements and the constraints imposed to the system. Although in the usefulness of SysML for modelling RF systems has been analysed, the migration of legacy Digital PHY system to a variant one has not been introduced and thoroughly proposed. Consequently, this case study focuses on the SysML modelling of the digital part of the NB PHY and especially its variants towards the migration to the UWB PHY.

7. SYSML MODELLING

The proposed development environment used for SysML modeling is the Eclipse IDE with the Papyrus plug-in. Eclipse/Papyrus environment is a flexible open source framework, and also an established platform with lots of support and plugins. Last but not least, an open source environment based methodology has potential for wider adoption that can be spread to SMEs due to no cost of the development environment.

Indicative part of the SysML diagrams, we developed, are depicted in the following figures.

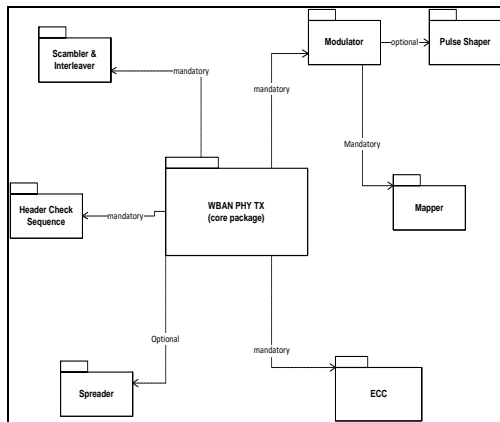


Figure 6. WBAN PHY TX package diagram

In Figure 6, a high-level package diagram of the WBAN PHY TX with all the *mandatory* and *optional* packages is depicted according to the methodology and the metamodel for package variability previously defined in this paper (Figures 1 and 2).

In Figure 7, three of the packages from Figure 6 are expanded with the corresponding SysML Block Definition Diagrams (BDDs). In these BDDs, the variability at block level is depicted according to the methodology and the metamodel previously defined (fig. 3 and fig. 4). *Mandatory*, *optional* and *alternative* are depicted through the proposed in this paper methodology.

Also, in Figure 7, a variant view where some of the *alternatives* are chosen. The grey boxes depict the choice of *Pulse Shaper* from the *Pulse Shaper* package as the *Single Pulse Shaper*. The dark grey boxes depict the choice from the *Mapper* package as the

(*ON-OFF Keying*) *Mapper* of $R=1/2$ ($R=1$ for *PHR*) $K=1$ & $M=2$ block. It should be highlighted that upon the variant view enforcement of a block (Figure 7), which has been previously defined as *optional* at the level of system description, is consequently becoming *mandatory*.

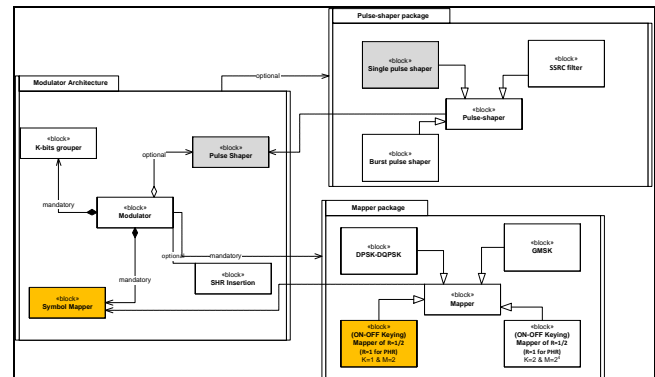


Figure 7. WBAN PHY TX package diagram with SysML Block Definition Diagrams (BDDs)

8. CONCLUSIONS AND FUTURE WORK

This paper has sought to show how SysML can be used, as part of the method described within, to allow developers to evolve variants of existing products, whilst still maintaining compliance with changing standards. This is illustrated by reference to specific industrial exemplars.

Whilst we believe that this approach illustrates that there is a promise for genuine industrial gains, we recognise that this is early work, and note that there is still much to be done in order for this approach to be developed fully. In particular, appropriate tools to assist the defined method that extend/modify the existing SysML specification need to be developed. Automated code generation towards the formation of an executable specification enriched by relevant signal processing libraries targeting to the simulation and evaluation of the system's performance, as a first step, needs to be developed. The second step that leads to high-level synthesis and final implementation of the system could employ the code generation and mapping of the SysML description to SystemC. In addition, the transition of current industrial business processes into new model-based paradigms needs to be formally defined. We also intend to apply our method to other application areas beyond eHealth standard compliant systems in order to evaluate its suitability and generality. Last but not least, we will undertake empirical studies to gauge the utility of the proposed method.

9. REFERENCES

- 1 IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications. In *IEEE Std 1901* (2010), 1-1586.
- 2 IEEE Standard for Local and Metropolitan Area Networks - Part 15.6: Wireless Body Area Networks. *IEEE Std 802.15.6-2012* (February 29, 2012), 1-271.
- 3 Bouabana-Tebibel, Thouraya, Rubin, Stuart Harvey, and Bennama, Miloud. Formal modeling with SysML. In *Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on* (2012), 340-347.
- 4 Kwak, Kyung Sup, Ullah, Sana, and Ullah, Niamat. An overview of IEEE 802.15. 6 standard. In *Applied Sciences in*

- Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on* (2010), 1-6.
- 5 Saggio, Giovanni, Cavallo, Pietro, Bianchi, Luigi, Quitadamo, Lucia R, and Giannini, Franco. UML model applied as a useful tool for Wireless Body Area Networks. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace \& Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on* (2009), 679-683.
 - 6 Group, Object Management. *Systems Modeling Language, SysML*. 2010.
 - 7 Roques, Pascal. SysML vs. UML 2: A Detailed Comparison. *MoDELS 2011* (2011), 16-21.
 - 8 What are Standards? Why are they important? *IEEE Standards Association, standards insight*, Inside the IEEE-SA, Standards at Work (Mar. 10, 2011).
 - 9 Meacham, Desmond J, Michael, James Bret, Shing, Man-Tak, and Voas, Jeffrey M. Standards interoperability: Applying software safety assurance standards to the evolution of legacy software. In *System of Systems Engineering, 2009. SoSE 2009. IEEE International Conference on* (2009), 1-8.
 - 10 INCOSE. *Systems Engineering Vision 2020, v.2.03*. 2007.
 - 11 Broy, Manfred, Kirstan, Sascha, Krcmar, Helmut, Sch, and Zimmermann, Jens. What is the benefit of a model-based design of embedded software systems in the car industry? *Software Design and Development: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications* (2013), 310.
 - 12 Chen, Jie and Patton, Ron J. *Robust model-based fault diagnosis for dynamic systems*. Springer Publishing Company, Incorporated, 2012.
 - 13 Towers, James. What is Model-Based Systems Engineering? *Z9, INCOSE UK* (February 2015).
 - 14 Grimm, Frank. Increasing the Reliability of Model-driven Software Family Engineering and Product Configuration for Automotive Controller Software. In *VaMoS* (2008), 33-41.
 - 15 Van Gorp, Jilles, Bosch, Jan, and Svahnberg, Mikael. On the notion of variability in software product lines. In *Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on* (2001), 45-54.
 - 16 Maga, CR and Jazdi, N. Survey, Approach and Examples of Modeling Variants in Industrial Automation. *Journal of Control Engineering and Applied Informatics*, 13, 1 (2011), 54-61.
 - 17 Friedenthal, Sanford, Moore, Alan, and Steiner, Rick. OMG Systems Modeling Language (OMG SysML™) Tutorial. In *INCOSE Intl. Symp* (2006).
 - 18 Haber, Arne, Rendel, Holger, Rumpe, Bernhard, Schaefer, Ina, and Van Der Linden, Frank. Hierarchical variability modeling for software architectures. In *Software Product Line Conference (SPLC), 2011 15th International* (2011), 150-159.
 - 19 Delligatti, Lenny. *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley, 2013.
 - 20 Holt, Jon and Perry, Simon. *SysML for systems engineering*. IET, 2008.
 - 21 Tar, Traver, Lara, and Cardona, Narc. Wireless body area networks for telemedicine applications. *Magazine Waves* (2009), 124-132.
 - 22 Kartsakli, Elli, Lalos, Aris S, Antonopoulos, Angelos, Tennina, Stefano, Renzo, Marco Di, Alonso, Luis, and Verikoukis, Christos. A Survey on M2M Systems for mHealth: A Wireless Communications Perspective. *Sensors*, 14, 10 (2014), 18009-18052.
 - 23 Chen, Mengyuan, Han, Jun, Fang, Dabin, Zou, Yao, and Zeng, Xiaoyang. An ultra low-power and area-efficient baseband processor for WBAN transmitter. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific* (2013), 1-4.
 - 24 Choi, BoKeun, Kim, ByungSoo, Lee, SangSeol, Wang, KyuYeul, Kim, YongJun, and Chung, DuckJin. Narrowband Physical Layer design for WBAN system. In *Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on* (2010), 154-157.
 - 25 Lafi, Sabeur, Champagne, Roger, Kouki, Ammar B, and Belzile, Jean. Modeling radio-frequency front-ends using sysml: A case study of a umts transceiver. *ACESMB 2008* (2008), 115.