AARHUS
UNIVERSITY
SCIENCE AND TECHNOLOGY

Energy-Aware System-Level Design of Cyber-Physical Systems

Department of Engineering

José Antonio Esparza Isasa

PhD Thesis

ISBN: 978-87-7507-322-1

DOI: http://dx.doi.org/10.7146/aul.14.13

Aarhus University Library

Energy-Aware System-Level Design of Cyber-Physical Systems

PhD Dissertation

José Antonio Esparza Isasa

April, 2015

AARHUS
UNIVERSITY
DEPARTMENT OF ENGINEERING

Energy-Aware System-Level Design of Cyber-Physical Systems

A Dissertation
Presented to the Faculty of Science and Technology
of Aarhus University
in Partial Fulfilment of the Requirements for the
PhD Degree

by
José Antonio Esparza Isasa
April 30th, 2015

Abstract

Cyber-Physical Systems (CPSs) are heterogeneous systems in which one or several computational cores interact with the physical environment. This interaction is typically performed through electromechanical elements such as sensors and actuators. Many CPSs operate as part of a network and some of them present a constrained energy budget (for example, they are battery powered). Examples of energy constrained CPSs could be a mobile robot, the nodes that compose a Body Area Network or a pacemaker. The heterogeneity present in the composition of CPSs together with the constrained energy availability makes these systems challenging to design. A way to tackle both complexity and costs is the application of abstract modelling and simulation. This thesis proposed the application of modelling at the system level, taking energy consumption in the different kinds of subsystems into consideration. By adopting this cross disciplinary approach to energy consumption it is possible to decrease it effectively. The results of this thesis are a number of modelling guidelines and tool improvements to support this kind of holistic analysis, covering energy consumption in electromechanical, computation and communication subsystems. From a methodological point of view these have been framed within a V-lifecycle. Finally, this approach has been demonstrated on two case studies from the medical domain enabling the exploration of alternative systems architectures and producing energy consumption estimates to conduct trade-off analysis.

Resumé

Cyber-fysiske systemer (CPSs) er heterogene systemer, hvor en eller flere computerenheder interagerer med de fysiske omgivelser. Denne interaktion sker typisk igennem sensorer og aktuatorer. Mange CPSs er tilkoblet et netværk, og nogle af disse er batteridrevet og har demed en begrænset energimængde til rådighed. Eksempler på CPSs med begrænset energimængde er en selvkørende robot eller trådløse og bærbare medicinske enheder. Forskelligheden af de komponenter der indgår i et CPS sammen med den begrænsede energimængde gør disse systemer komplekse og udfordrende at udvikle. En måde at håndtere kombinationen af kompleksiteten og den begrænsede energimængde på, er at anvende modellering og simulering tidligt i udviklingsforløbet.

Denne afhandling foreslår at anvende modellering på systemniveau, og at inddrage energiforbruget i de forskellige delsystemer i modelleringen. Ved at anvende denne tværfaglige tilgang mht. energiforbruget er det muligt at sænke forbruget effektivt. Resultatet af denne afhandling er dels en række retningslinjer for modelleringen, og dels forbedring af værktøjer til at understøtte denne type holistisk analyse, der omhandler energiforbruget i elektromekaniske-, computer- og kommunikationsdelsystemer. Fra et metodemæssigt synspunkt er disse retningslinier udformet ift. en V-udviklingsmodel. Til slut er denne tilgang blevet anvendt og afprøvet i to case-studier fra det medicinske domæne. I case-studierne er der undersøgt alternative systemarkitekturer og beregnet energiforbrugsestimater til brug ved trade-off analyser.

Acknowledgments

First of all I would like to thank my supervisors Professor Peter Gorm Larsen and Senior Associate Professor Finn Overgaard Hansen. I met them for the first time when I started my Master studies back in 2010 and, since then, I have had ample opportunity to learn from their professional and research experience. This has had a profound impact on the quality of my work, made me a better engineer and contributed to my personal development. To both of them thank you.

Throughout my PhD studies at Aarhus University I have had the pleasure to meet great colleagues that have contributed to the development of my work in various ways. Peter H. Mikkelsen has provided very good advice on hardware related topics in a very sharp and clear way, making complex hardware concepts easier to understand. Troels Fedder Jensen has been giving me advice on software development, collaborating on prototype development and helped in countless occasions in many other ways. Michael Alrøe provided expert insight on embedded hardware and software issues, especially at the beginning of this PhD. Qi Zhang has shared her experiences on the research world as well as her expertise and has always been up for interesting conversations. To all these colleagues and many others, thank you.

Peter W. V. Tran-Jørgensen, a fellow PhD student and good friend, deserves special mention in this section. I met him at the beginning of my PhD, we have gone to conferences, participated in courses and conducted joint work together. He has a helpful, positive and collaborative attitude besides being an excellent software engineer and researcher.

As part of this PhD thesis I had to step into areas that were completely new to me, such as mechanical modelling. This was a challenge in which I received help from several people, especially from fellow PhD student Martin Christiansen and from my good friend Andreas Thomasen, for whom traditional mechanics results trivially from natural laws, given the fact that he is a quantum physicist. Nick Battle and Joey Coleman reviewed preliminary versions of the material presented in this thesis with patience and interest, providing useful advice regarding its presentation. Thanks to you all for your help and contributions.

I would like to thank my co-authors Claus Ballegaard Nielsen for his collaboration and helpful discussions in the modelling field as well as Timo Vuroela and Tomi Hakkala from Tampere University of Technology, for their innovative input during the e-stocking project.

This section would not be complete without thanking my colleagues from the Software Engineering group at the Department of Engineering. They have a truly helpful attitude and are always willing to share their views, expertise and to provide useful feedback. All of them are very competent professionals and I am looking forward to our continued collaboration in upcoming projects.

The support that my family has given me during the PhD studies as well as during my education has been outstanding. They have always encouraged me to continue studying, expand my knowledge and strive to become better. My being where I am today is in no small amount thanks to their encouragement.

Finally, I would like to thank my fiancée, Julija Skolina. She has been a great support through all my years of studying, coped with piles of electronics, prototypes and papers in the middle of the living room and showing me a good time when research sometimes became too much. We have shared together every single day of these PhD studies and I have always been able to count on her exceptional support and understanding. Especially to her, thank you.

José Antonio Esparza Isasa
Aarhus, April 2015

Contents

Part I

Summary

1

Introduction

"After cruising more than a 1000 million Km, away from Earth and after a hibernation of almost three years, the European Space Agency space probe Rosetta autonomously woke up, warmed up its navigation instruments, stabilized its course, beamed its antenna and contacted mission control."

European Space Agency, 20th of January 2014 [2]

1.1 Introduction to the field of research

The concept of Cyber Physical Systems was introduced by Helen Gill from the National Science Foundation in 2006, who described these kinds of systems as follows:

"Cyber-Physical Systems are physical, biological and engineered systems whose operations are integrated, monitored and/or controlled by a computational core. Components are networked at every scale. Computing is deeply embedded into every physical component, possible even into materials. The computational core is an embedded system, usually demands real-time response and is most often distributed"[1]

Many other definitions exist in the literature but almost all authors explicitly highlight the importance of communicating computational cores and their interaction with the physical world [96, 73]. The concept of CPSs is very broad and many systems of different kinds and nature could be considered a CPS. Examples of CPS are Heating Ventilation and Air Conditioning

[1] Although no original source has been found for this statement, it is commonly accepted in the research community and attributed to Helen Gill. See [73].

(HVAC) systems, on-board car computers or pacemakers among many others. This gives an idea of the complexity, scale and heterogeneity of CPSs. Typically systems like these are engineered by multidisciplinary teams of engineers, that are composed of experts within different fields such as mechanical, computer and electrical engineering.

One of the challenges faced by CPSs development teams is that these systems are expected to work in a dependable, safe and secure manner [96] and very often under a limited energy budget, requiring them to be energy efficient [6]. For example, the Rosetta space probe mentioned earlier could be considered a CPS that operates depending on the energy availability. This probe entered a "deep sleep" mode where most of its instruments were deactivated when it could not use its solar panels and resumed its normal, active, operational mode as soon as it was close enough to the Sun again [2]. Such "smart energy management" has made it possible to adjust the behaviour of the probe depending on the operational conditions and ultimately, that the probe is still working today. A second relevant example of a CPS with a limited energy budget and used widely today is a pacemaker. Pacemakers have undergone a dramatic evolution with regard to their size, weight, functionality and battery lifetime [84]. The first pacemakers had an expected battery lifetime of around 2 years while with the current technologies it is possible to reach up to 8 years of continuous, reliable operation [92]. Moreover, energy harvesting techniques have also been proposed so pacemakers are powered from vibrations [60] or thermal gradients [5].

These systems are very complex and need a systematic engineering approach to their design. A possible approach is the creation of models and conducting different kinds of simulations and measurements in order to address the engineering challenges behind them. Such an approach has traditionally been applied in other (more mature) engineering disciplines. For example, in the civil engineering field modelling and abstraction are routinely used to analyze buildings and constructions. In Figure 1.1 for example, Gaudi's sand bag model of La Sagrada Familia, Barcelona, Spain is presented. This was developed in the early 20^{th} century and it is an abstract model of the cathedral structure, that represents the loads with sandbags. Different loads could be modelled by adding or removing sand from the bags and therefore it was possible to analyze the impact of different design decisions in the overall cathedral structure.

Many authors agree that the CPSs discipline has to adopt the successful experiences from more mature engineering disciplines and incorporate them into its body of knowledge in the creation of its scientific foundation. One of the key elements there is the application of modelling and simulation to analyze and design systems [73, 62], like Gaudi did during the design of La Sagrada Familia.

Figure 1.1: Gaudi's Sandbag model of "La Sagrada Familia" Source: `http://commons.wikimedia.org/`

1.2 Motivation

Since the definition of the CPSs field by Helen Gill in 2006, its importance has been rising and this has had a considerable impact on the research agendas of the US and EU scientific organizations. This has been reflected on the research budget allocation from the National Science Foundation (NSF) and the European Commission (EC).

In the US the White House Office of Science and Technology included Cyber-Physical Systems research under the category "Advanced manufacturing and industries of the future" and identified it as a "multi-agency R&D top priority" [20]. Consequently the NSF is actively funding research projects in this area with specific calls for CPSs [34].

On the European side, the EC in its new research programme Horizon 2020 identifies a "new generation of components and systems" that refers specifically to embedded systems and CPSs as "two of the main challenges to reinforce and expand Europe's leading industrial position" [26, 28]. Consequently the EC is actively funding numerous research projects related to CPSs. In a similar manner low-power computing is among the EC research priorities and funded with specific calls under the category "Advanced Computing" [27, 28]. The ARTEMIS initiative, who plays a key role in Embedded and CPSs research at the European level, targets in their Strategic Research Agenda model-based engineering of CPSs in what they call "virtual engineering of CPSs". Additionally they also identify the need for better "computer platforms and advanced energy management in CPS" [40]. Additionally, the Electronic Components and Systems for European Leadership (ECSEL) consortium identifies in its Strategic Research and Innovation agenda CPS as one of the essential capabilities that will facilitate the progress in key areas for societal development such as energy, health and production [23]. Finally, the creation of models and methods for System-Level evaluation, verification and validation of CPS has been explicitly formulated as one of the strategic opportunities for the development of CPS [105].

These points show the relevance of the CPS discipline in the current research landscape as well as the potential impact in today's engineering practice in the the industrial sector. This, combined with the ubiquity of CPS in the existing devices, infrastructures and services, make the area of Low-Power CPS an open challenge to be addressed in order to enable further developments.

1.3 Research methods

The research methods applied in this PhD thesis are composed of four main phases that have been executed sequentially. This method has been applied separately for the main areas of this PhD thesis (computation, communication and electromechanical energy consumption modelling[2]). For each area the method has been applied in an iterative manner until the research objectives have been achieved, in a similar way to the Bohem Spiral Model [10]. The different phases of the research method and their connectors are shown in Figure 1.2.

The phases that compose this model are:

Concept Formulation: In this phase the model-based approach to energy consumption for the particular area under study is formulated. This is

[2] This division will become more clear in the following chapters.

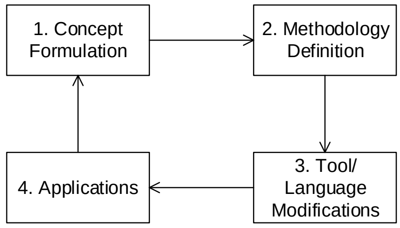


Figure 1.2: Overview of the research methods applied in this PhD.

carried out after reviewing the state of the art and identifying how the current challenges could be addressed through modelling and simulation.

Methodology Definition: Once the concept has been formulated, this phase examines how well the baseline technologies perform when modelling subsystems that fall under the area under consideration. In case modelling or methodological limitations are identified these are marked as research goals that will have to be addressed.

Tool/Language Modifications: If the baseline technologies are limited with regard to the analysis they can perform in terms of power and energy consumption, these are addressed in this phase.

Applications: The modifications carried out in the previous phase are applied in a manageable case study or simple applications. This allows testing of how well these modifications are performing before using them in more complex case studies.

After using these four phases individually over the different main areas addressed in this work, the methodological and tool extensions have been applied in two different real case studies in order to assess their performance. An overview of how this approach has been used in the thesis is presented in Figure 1.3.

1.4 Research objectives

This PhD thesis aims to addressing some of the methodological and tool-related challenges behind the creation of CPSs with a limited energy budget.

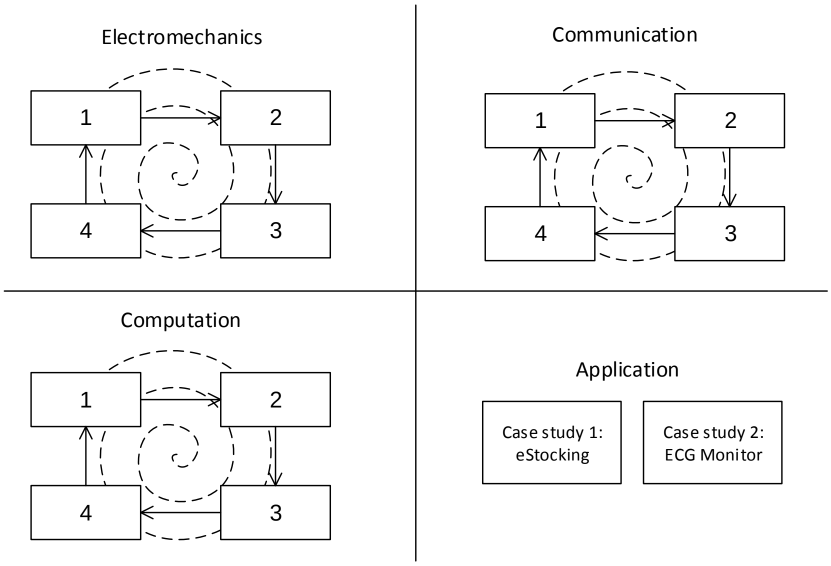


Figure 1.3: Overview of the overall methodology used in this thesis.

It is the central proposition of this PhD that:

> Non-trivial CPS need to be engineered by multidisciplinary teams. Additionally CPS often have critical energy consumption requirements. As a result, non-trivial energy constrained CPS are composed of subsystems that belong to different domains, that have to be addressed by multiple disciplines, yet they inevitably share a cross-cutting concern, since all of them use energy. Furthermore, a design decision in a subsystem belonging to one domain may impact subsystems in other domains, both in functional terms and very possibly from the power and therefore energy consumption point of view. This is due to the tight coupling between physical subsystems and computational and communication ones. The engineers developing CPSs need to be able to cope with these kinds of challenges by having a System-Level approach to the design of CPSs.

The hypothesis of this PhD is that:

> A model-driven engineering approach, that applies heterogeneous modelling techniques combined with partial prototyping, will enable a holistic approach that is needed to take energy consumption into account during the System-Level design of CPSs.

It is our hope that the results of this PhD project will inspire other researchers and practitioners working on CPSs with a limited energy budget. Additionally we hope that the application of the methodological and modelling principles presented in this work to two real case studies will give an idea of the advantages that this model driven approach can bring in an industrial setting.

1.5 Evaluation criteria

The work conducted in this thesis has produced a number of research contributions that are evaluated individually against the different dimensions presented below. The contributions will be presented in Chapters 3 and 4 and their evaluation is conducted in Chapter 5. The evaluation dimensions are:

Improvement of modelling methods and tools: addresses energy and power related aspects in CPSs design that will require the extension and/or the proposal of methodological guidelines.

Better System-Level representation of CPS: helps to achieve a more complete understanding of the CPS under study. Additionally it should enable the practitioners to contextualize and to evaluate the impact on power and energy consumption of different design decisions across different subsystems.

Collaboration between engineering disciplines: takes into consideration the different engineering disciplines involved in the inception, realization, validation and verification of CPS.

Application to real case studies: demonstrates the applicability of the contributions to the development of CPS in a practical setting. These should not be limited to toy examples but should also perform well on realistic systems. These can be of small to medium size in the beginning but still cases that could be found in a real industrial context.

Shortening of development times and reduction of costs: evaluates the reduction of development times by enabling the co-design of subsystems, potentially reducing manual labor and enabling a faster and higher coverage when evaluating the design space.

Score	Description
0	Contribution is not relevant to this dimension.
1-3	Contribution is relevant to this dimension and assessed depending on its degree of completeness and the extent of its results.
4	Contribution is relevant to this dimensions, well formulated and covering existing needs. It has the potential of being applicable in other fields or taken as a reference for further developments.
5	Contribution is complete and relevant to this dimension. It can be applicable in other fields and taken as a reference in current state of the art industrial project and academic research.

Table 1.1: Overview of the different scores.

Different contributions will score differently for these different evaluation dimensions but hopefully they will complement each other so the collection of contributions will demonstrate overall progress supporting the hypothesis of this thesis. A score system of 0 to 5 has been used to the evaluate the performance of each contribution in each different dimension. The criteria used to assign the different scores are presented in Table 1.1. In order to present the assessment of each individual contribution as well as the overall one radar charts as the one shown in Figure 1.4 will be used. For each contribution one graph is used and for each dimension an axis is assigned. The polygon resulting from connecting the scores for each dimension represents the overall assessment of the contribution.

1.6 Published work

The research results achieved throughout this PhD project have resulted in several publications. Most of the publications are related to the core work conducted in this PhD and are listed in subsection 1.6.1. Two additional publications specific to the Ambient Assisted Living (AAL) project “e-Stockings” are listed in subsection 1.6.2. Finally, in subsection 1.6.3 other publications not related to this project are listed.

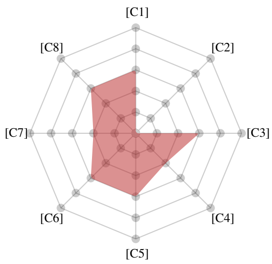


Figure 1.4: Example radar chart.

1.6.1 Core publications

[P54] José Antonio Esparza Isasa and Peter Gorm Larsen and Finn Overgaard Hansen *A Holistic Approach to Energy-Aware Design of Cyber-Physical Systems*. Submitted to the International Journal of Embedded Systems.

[P49] José Antonio Esparza Isasa, Peter Gorm Larsen and Finn Overgaard Hansen *Embedded Systems Energy Consumption Analysis Through Co-modelling and Simulation*. Proceedings of the International Conference on Modeling and Simulation (ICMS 2013) June 2013.

[P52] José Antonio Esparza Isasa and Peter Gorm Larsen *Modelling Different CPU Power States in VDM-RT*. Proceedings of the 11th Overture Workshop September 2013

[P51] José Antonio Esparza Isasa, Peter W.V. Jørgensen and Peter Gorm Larsen *Hardware In the Loop for VDM-Real Time Modelling of Embedded Systems*. 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2014) January 2014

[P50] José Antonio Esparza Isasa, Peter W.V. Jørgensen and Claus Ballegaard Nielsen *Modelling Energy Consumption in Embedded Systems with VDM-RT*. Proceedings of State Machines, Alloy, B, VDM, and Z 2014 (ABZ 2014), April 2014.

[P55] José Antonio Esparza Isasa, Peter Gorm Larsen and Finn Overgaard Hansen *Energy-Aware Design of Embedded Software through Modelling and Simulation*. Proceedings of the SYRCoSE 2014 Symposium. May 2014.

[P56] José Antonio Esparza Isasa, Peter Gorm Larsen and Finn Overgaard Hansen *Energy-Aware Model-Driven Development of a Wearable Health Care Device*. 4th Symposium FHIES/SEHC June 2014. In Press.

1.6.2 e-Stocking project publications

[P57] Troels Fedder Jensen and Finn Overgaard Hansen and José Antonio Esparza et al. *ICT-Enabled Medical Compression Stocking for Treatment of Leg-Venous Insufficiency*. 7th International Conference on Biomedical Electronics and Devices (BIODEVICES 2014), March 2014.

[P44] Finn Overgaard Hansen and Troels Fedder Jensen and José Antonio Esparza *Distributed ICT Architecture for Developing, Configuring and Monitoring Mobile Embedded Healthcare Systems.* International Conference on Health Informatics (HEALTHINF 2014), March 2014.

1.6.3 Publications outside the scope of this PhD

[P53] José Antonio Esparza Isasa and Peter Gorm Larsen and Kim Bjerge *Supporting the Partitioning Process in Hardware/Software Co-design with VDM-RT*. Proceedings of the 10th Overture Workshop, August 2012.

1.7 Outline and reading guide

This PhD thesis is divided in two parts. Part I provides the necessary background to understand this PhD project, summarizes the contributions and evaluates them. Part II includes a selection of the papers published by the author in cooperation with other researchers that are relevant to this PhD thesis.

Part I is composed of five chapters. After the current introductory one, Chapter 2 presents the information needed to understand the remainder of this thesis. Chapter 3 presents one of the core results of this PhD thesis, a System-Level Model-Based Approach to the design of energy-constrained CPSs. This chapter provides methodological additions and guidelines to study through modelling some of the energy-related challenges present in CPSs. This chapter is based on the publications [P49, P52, P51, P50, P55]. Chapter 4 presents the application of these methods to two case studies: a compression stocking to treat leg-venous insufficiency and an heart monitor. This chapter is based on the publications [P56, P57]. The contributions of this PhD project are presented throughout Chapters 3 and 4. Each contribution is introduced in a frame standing out from the main body of the text as shown below.

Contribution #. Description of the contribution.

Finally, Chapter 5 summarizes this thesis and assesses the work conducted during the PhD project.

Part II contains the publications [P49, P52, P51, P50, P55, P56, P57, P44, P54].

2

Background

2.1 Cyber-Physical Systems

Cyber-Physical Systems (CPSs) are complex systems composed of Electro-Mechanical Subsystems (EMS) and embedded computer subsystems. The EMS typically feature sensors and actuators so the CPS can interact with the physical environment. In order to control them, CPSs integrate embedded computer systems [72]. Additionally, CPSs can operate as part of a network and therefore integrate a communication subsystem. This heterogeneous composition can be captured in the SysML block diagram shown in Figure 2.1. In this diagram a hypothetical CPS system is shown which is composed of: a computation subsystem that consist of embedded hardware as well as embedded software, of a communication subsystem that consist of communication logic that implements communication protocols (in either hardware or software) and the Frontend and also electromechanical subsystems controlled from the computation side. The CPS also contains an energy subsystem. This subsystem can integrate ”limited” energy sources such as batteries or have connections to ”unlimited” sources, considering as such a power outlet.

There are many different kinds of CPSs and therefore it is difficult to provide a complete overview here, however the following examples give an idea of the variety that can be found within the CPS field:

- Wearable systems such as sports wristbands or different kinds of medical devices [14]. These devices typically perform monitoring of different physiological parameters and are battery powered. Some of them can perform treatment of specific medical conditions. These kinds of devices are gaining popularity at the moment thanks to the appearance of enabling technologies in sensing, computing and communication [109]. Furthermore the concept of having these devices conforming a Body Area Network is emerging and gaining popularity within the CPSs and healthcare community [13]. Ensuring energy autonomy and a correct and reliable application is especially critical in this kind of devices.

- Mobile platforms such as drones (UAVs), agricultural robots or industrial robots [95]. These devices interact with the physical environment in order to perform a variety of tasks that have been programmed previously or that are issued dynamically. Some of the open challenges in the development of this kind of devices are precision and safety in dynamically changing environments and energy consumption.

- Heating Ventilating and Air Conditioning (HVAC) systems [77] or Data Centers [91] are systems that at first sight seem to be completely unrelated to the CPS field and more to civil engineering or computer science respectively. However, in both cases the thermal management in both kinds of infrastructures is a problem in which both control software, physical interaction with the environment and energy management are involved.

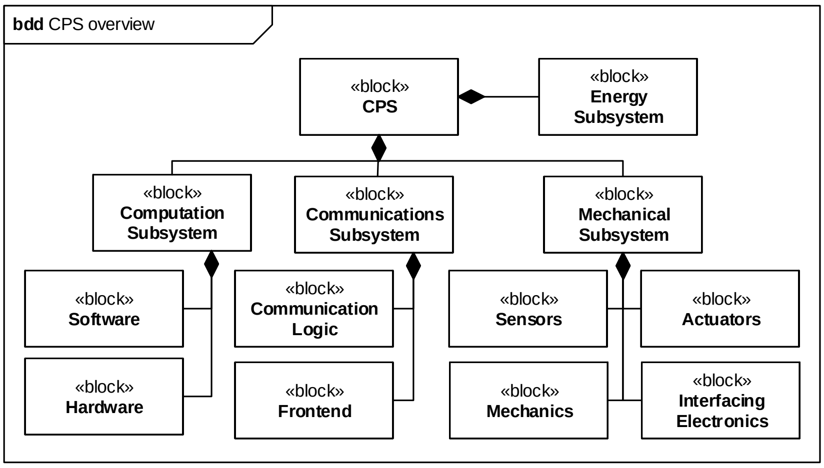


Figure 2.1: Structure of a hypothetical CPS represented in a SysML block diagram.

CPSs are typically developed by several different engineering disciplines working together due to their heterogeneous composition. This requires that the disciplines involved in the development of the CPS communicate and cooperate during the inception, creation and, especially, integration of the subsystems that compose a CPS. In this case the application of best practices of systems engineering and methods and tools that help to tackle this complexity are a key to success.

Some of the additional design challenges posed by CPS are Safety, Sustainability and Security, and these are known as the S3 properties as defined in [6]. In order to satisfy them, especially the sustainability aspect, there is a need for better algorithms, analysis tools, methods and modelling techniques [43].

2.2 Power and energy consumption

2.2.1 Electrical power and energy consumption formulation

In order to calculate the power consumption of the systems under study at a specific point of time, the general formulation of power consumption in electrical DC systems has been used. This is defined as:

$$P \text{ [Watts]} = V \text{ [Volts]} \times I \text{ [Amperes]} \tag{2.1}$$

Based on the power consumption the energy consumption is calculated as its integral over a certain period of time:

$$E \text{ [Joules]} = \int_{t_0}^{t_1} P \text{ [Watts]} \, \mathrm{d}t \tag{2.2}$$

2.2.2 Power consumption reduction techniques for CPSs

Regarding power consumption optimization in electrically powered systems there are many techniques that can be used and this depends on the nature of the system under study. While some techniques are used across multiple domains other are restricted to specific ones. The most relevant power saving techniques that are referred to in this thesis are:

Power Gating: this is a general technique that can be used across several domains. It consist on the selective deactivation of components that are not used in the system by disconnecting their supply line. This reduces the current draw to the leakage current of the components performing the gating, which is orders of magnitude below the current draw when the system is operational. This can be applied through Application Programming Interfaces (APIs) in many software controlled devices such as CPUs or Communication interfaces. In principle any kind of device can be gated by cutting its supply line through a MOSFET or specialized low-current leakage switch loads [47] such as [48].

Dynamic-Frequency Voltage Scaling (DFVS): this is a technique that is typically used in the computation domain. It consist of the dynamic adjustment of operational voltage and/or frequency depending on the load of the processor and they are normally applied together at the CPU architectural level. This has a direct impact on the CPU power consumption. At the application level typically it is possible to dynamically vary the CPU operational frequency through an API. A presentation of these techniques together with a description of how they are applied in concrete hardware platforms can be found in [100].

Sleeping States: this technique is typically used in the computation domain and consist of placing the CPU in a low power consuming state when it is idle and does not have to perform any computation. Additionally modern microcontrollers incorporate the necessary facilities to perform power gating over their different peripherals so not only the CPU saves power. Typically there are different kinds of Sleeping States that define the capacity of the microcontroller to react to events (both external and internal). For example ARM cores typically feature several sleep modes such as Deep Sleep with a very low power consumption but at the price of longer wake-up times. This can lead to restrictions if they have to react quickly to certain events. Additional details on these techniques for the ARM implementation provided in the platform PSoC5[1] can be found in [19].

Modem Operational States: this technique is applied in the communications domain where communication interfaces can have several operational states. For example, modern 802.15.4 communication interfaces feature states such as transmission, listening or low-power listening in which they communicate or remain in a very low power consuming state awaiting for an incoming request. This resembles the principle behind

[1] PSoC stands for Programmable System on Chip and it is a platform manufactured by Cypress Semiconductors. This platform is a reconfigurable integrated circuit that contains both analog and digital components. Further details can be found in: `http://www.cypress.com/`

the Sleeping States technique described above and in general the idea of multiple operational states is applicable in subsystems of any kind and at different levels of abstraction. More details on this can be found on [102].

A comprehensive review of techniques to study power consumption in computation can be found in the survey paper [107]. In a similar manner, further information on energy-efficient wireless communication can be found in [30]. Further details on the different power consumption issues involving computing and communication aspects in the wireless sensor networks context can be found in [65]. From the mechanical point of view there are many techniques that could be applied depending on the mechanical system under consideration, being most of them very specific. Some of them are rather generic guidelines such as reducing friction as much as possible between moving elements or right-sizing motion control systems [18]. These techniques are beyond the scope of this work and therefore no further details will be provided.

2.2.3 Power and energy consumption as design parameters

As it can be seen in equations 2.1 and 2.2 power and energy consumption are two different concepts. This has an impact when taking power and energy consumption into consideration in order to design systems. As explained by Unsal et al. in [107]:

*"**Power and energy efficiency are separate design goals.** Energy is the integral of power consumption over a time period. A power-efficient design might very aggressively decrease the [CPU] clock rate, but this might not be an energy efficient design since the performance of the application might degrade to such a degree that the actual energy consumption increases."*

Following the same reasoning Unsal et al. also pointed out that a power-constrained application is different from an energy constrained one [107]. Take as an example a wireless sensor node powered by a radioactive power source[2], such as the radioisotope Nickel-63. This isotope could potentially power a sensor node until it reaches its half-life after 100 years [76]. Researchers have been able to power a sensor node and make it transmitt a 5mW RF pulse every three minutes based on this radioactive based energy harvesting technique [1]. One could claim that the energy availability in this device is not a problem due to the relatively long half-life of the Nickel, however, due to the small amount of isotope integrated in the node as well as the low efficiency of the energy harvesting process, the actual electrical power available for the device to use is very small. This implies that the designers of such a device should adjust its operation taking power consumption into consideration (and not energy).

2.2.4 Power and energy Awareness

The term "aware", defined as "having knowledge or realization of; being conscious of" can be seen frequently appended to the terms "power" or "energy", resulting on the qualifiers power-aware and energy-aware (see [41] or [67]). Here it is important to clarify the following concepts:

- A system that is power-aware or energy-aware can dynamically modify its behavior depending on the power or energy availability. As explained before, the system can take power or energy consumption into consideration in isolation as design parameters (or both) [107].
- A design process or methodology that is power and/or energy-aware, like the one proposed in this thesis, is able to represent power and energy consumption and take them into consideration when designing and realizing the system under study.

In the remainder of this thesis the term *energy* will be used whenever there is a possibility of choosing between power or energy. The term *power* will be used when there is a specific need to refer to it.

2.3 Modelling methods and tools

Throughout the work conducted in this thesis a number of modelling methods and tools have been used to represent different subsystems present in CPSs. The languages discussed below have been used primarily to: A) facilitate the graphical and unambiguous representation of systems and models and B) create models to conduct analysis of different CPS aspects.

In order to create graphical models the Unified Modelling Language (UML) [85] and System Modelling Language (SysML) [104] have been used. UML

[2] Typically, an energy harvesting module based on a radioisotope works by collecting the emitted particles by the isotope over a surface that mechanically stimulates a piezoelectric element. This mechanism is known as energy converter and where the efficiency challenge lies, since current solutions can achieve an efficiency of 2-3% [68].

is a software modelling notation that allows the representation of structural, dynamic and deployment aspects of software systems. It is a graphical language and some tools offer code generation possibilities (mainly stub generation). UML is used through this thesis to represent architectural and deployment aspects of models developed with other, non-graphical notations. SysML extends UML so it is possible to represent other kinds of subsystems that are related to software. It incorporates new ways of representing a system's structure through block and internal block definition diagrams (BDD and IBD respectively). BDDs show the constituent elements of the system paying special attention to their hierarchy. IBDs show how the different constituent blocks that conform a block are instantiated and connected through the application of ports. SysML also introduces new graphical representations for requirements and physical behaviour [99]. In this thesis SysML is used to represent CPSs structure through BDDs and IBDs.

In order to create more formal representations as well as to conduct analysis and simulation the VDM-RT notation and the 20-sim platform have been used. These technologies can describe the system through Discrete Event (DE) or Continuous Time (CT) representations respectively. VDM-RT [71] is an extension to the object-oriented software modelling language VDM++ [32] and enables the modelling of real-time embedded control software. VDM-RT models can be created and executed in the Overture[3] tool [69]. This language incorporates the abstraction **`CPU`** that represents an execution environment in which parts of the model can be deployed. CPUs can be configured in terms of frequency and scheduling policy since they also incorporate a real-time operating system layer. Logical functionality running in VDM-RT CPUs can represent single or multi-threaded software implementations as well as dedicated hardware blocks depending on how they are configured [P53]. The VDM-RT CPUs only have one operational state, meaning that they are active (and therefore consuming energy) all the time. VDM-RT also incorporates the abstraction **`BUS`** that allows the **`CPU`** processing nodes to communicate. This abstraction can be used to represent point-to-point communication between CPUs in a static way. An example of these distribution aspects is shown in Figure 2.2, where two software components represented in VDM-RT are deployed in two different simulated CPUs that communicate via a bus.

[3] Overture's official website is: `http://overturetool.org/`

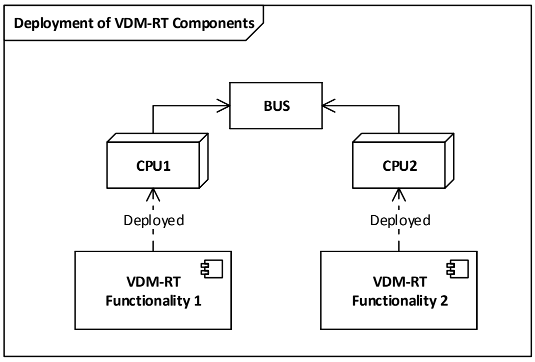


Figure 2.2: VDM-RT models deployed in CPUs.

In order to represent physical system aspects the CT simulation platform 20-Sim[4] is used. This platform is capable of representing electrical and mechanical systems among others. Modelling can be done by using bond graphs, iconic diagrams or differential equations. Standalone 20-Sim models can be created and executed in the 20-Sim tool [17, 63, 108]. An example model is presented in Figure 2.3. This is a model included in the platform examples package that represents an EMS composed of a servo motor, a mechanical transmission and its connection to a load. The position of the load is read through a sensor and this is fed into a PID controller [112], that operates the motor until its position matches the target setpoint.

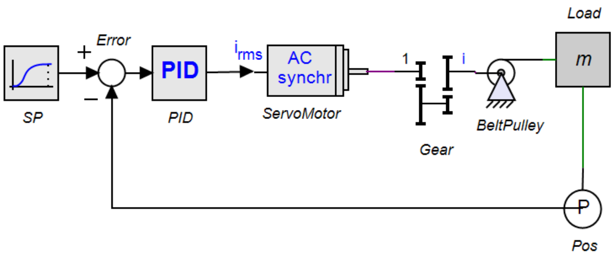


Figure 2.3: 20-sim model created using signal and iconic components.

[4] 20-Sim's official website is: `http://www.20sim.com/`

The Crescendo tool[5] is used in order to co-simulate models created using the VDM-RT notation and the 20-Sim platform [33]. This tool was originally developed during the DESTECS project[6] [11] and it controls a co-simulation using Overture and 20-Sim. The tool provides a common notion of time to the VDM-RT interpreter and the 20-Sim equation solver to synchronize the parallel execution of DE and CT models, now considered a single co-model. Additionally a number of variables are defined in a contract, which specifies the interfaces between the models in terms of controlled and monitored variables. Methodological guidelines on how to use the tool in order to represent CPSs can be found in [12, 33].

2.4 Modelling and analysis of energy consumption

Extensive previous work exists related to the application of models to capture energy consumption in systems with the purpose of reducing it. These techniques are very common especially at the computational and communications domain and are applied at different levels of abstraction. At the transistor and IC design level many design principles exists but these are out of the scope of this thesis.

At the computation side energy consumption had traditionally been addressed at the device level until the mid 90's [107] when it saw its expansion to higher levels of abstraction [75]. Going up in abstraction layers implied jumping from the electronic engineering to the computer engineering field, where a number of abstraction layers are present. Previous work can be found regarding modelling and profiling applied at different layers [90, 101] and also related to peripherals utilization [66, 87]. At higher levels of abstraction and involving software related aspects work has been conducted on modelling operating systems performance [7] and services [86]. Work is being conducted in order to raise the awareness of the embedded software developer to take energy consumption into consideration when creating software, based on what is called "energy transparency" [24]. This is possible through the application of static analysis of energy consumption. An example is the work conducted in the ENTRA project[7]. Under this approach, an Instruction Set Architecture is characterized based on experimental measurements [61]. Using that model it is possible to produce energy consumption estimates

[5] Crescendo official website is: `http://crescendotool.org/`
[6] DESTECS official website is: `http://www.destecs.org/`
[7] ENTRA official website is: `http://entraproject.eu//`

of software through the analysis of its Intermediate Representation at the compiler level [42]. This approach has been generalized so it is possible to take different configurations and inputs into consideration as parameters for simulations [74].

Energy consumption in computation has been studied in connection to specific computational aspects such as temperature [103] or reliability. In [38] an energy/reliability trade-off analysis is conducted based on a probabilistic model. As a result of this it is possible to find a reliable and energy efficient system implementation. This approach is primarily focused on scheduleability and application mapping. This analysis is more rigorous than what is possible to conduct with the VDM-RT modelling language and it could be used as a complimentary one to determine optimal application mappings.

The development of Wireless Sensor Networks [21] resulted in the creation of small embedded devices with both communication and computational capabilities. This extended the concept of System-Level to include both computation and communication aspects and therefore consider energy consumption in both areas [102].

Extensive work exists at different layers in the communication side ranging from specific physical models for concrete technologies such as BLE [25], network protocols [110, 111, 80] or focusing on the wireless node as an individual system and studying its individual energy consumption [89, 114]. There are specialized network simulators such as OMNET [16] that have been extended for taking energy consumption at the node level into consideration while enabling the simulation of network related aspects [29].

An additional perspective on this is the study of the energy source itself, by developing models of the battery powering the system [103, 97, 113]. Furthermore, work has been conducted on co-simulating communication aspects together with energy harvesting issues [22].

2.5 Systems engineering lifecycles

The development of systems is typically organized in lifecycles that span from the very beginning with the system inception until the final stages with the system acceptance. This lifecycle can be represented in several ways, the most commonly accepted being the V-model. This model structures the creation of systems in several phases that are matched with verification and validation phases. At the bottom of the V the actual realization is conducted. The naming of these phases vary depending on the organization proposing the V-model and normally companies tailor it so it suits their needs. In Figure 2.4

a simplified version of the V-model proposed by INCOSE [46] is presented. The creation of a system starts with an inception phase where the system concept is defined and the user requirements are captured. This is followed by two phases in which the system and the subsystems requirements are specified. After that the subsystems architectures are designed and finally realized. Following the realization, a number of Quality Assurance (QA) phases that form the right hand side of the V are carried out. These are matched with concrete parts on the left side of the V as shown in Figure 2.4.

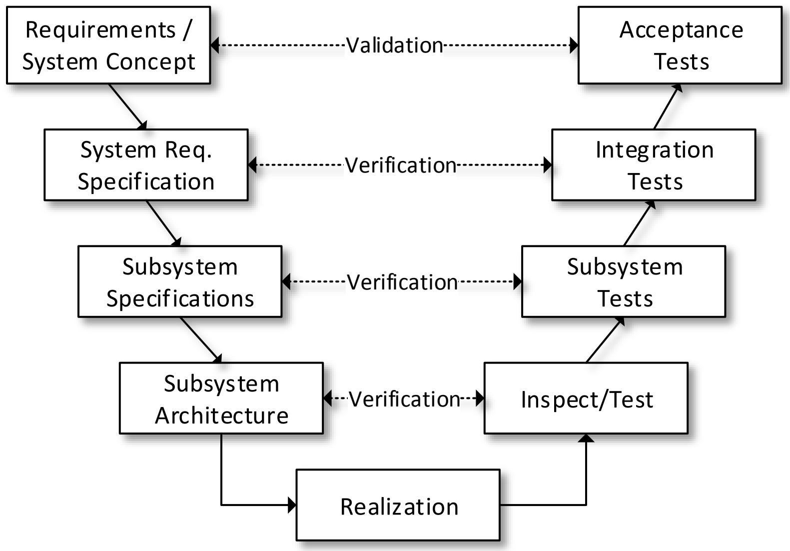


Figure 2.4: Simplification of the V-model proposed in [46].

The lifecycle described above is typically document intensive, meaning that it requires the production of requirements, design and QA documents for most of the phases. An alternative version to this is a more model-based lifecycle as shown in Figure 2.5, in which models of different kinds are used to support and document the different activities carried out in the phases. For example the IBM proposes Harmony, which is a methodology that could be framed within a V-model, uses SysML and UML to specify and represent systems [45]. However this methodology is still very focused around software development and not suitable for the development of entire CPSs. A step further in the model-driven engineering approach is a complete, model-based lifecycle, in which on the left side of the V the models are used to specify, analyze and explore the design space and on the right side of the V models, the analysis conducted over them and over the system/subsystems realizations can be used to conduct QA and gain confidence on the system produced. An overview of this lifecycle is presented in Figure 2.5. Such an approach has been already proposed for software-based systems using the use cases technique and the VDM notations [70]. While valid for embedded software systems and, to a limited extent, embedded hardware systems this lifecyle lacks methodological guidelines to address CPSs. This is in part complemented by the approach proposed in [33] where guidelines to addressed both cyber and physical parts of CPSs and model them using VDM-RT and 20-Sim respectively are proposed. Following a similar approach the CT modelling environment Modellica [36, 35] has been used to capture requirements in a clear and unambiguous manner and to facilitate their traceability [106] down to the implementation level. Additionally, this approach incorporates a systematic way to conduct QA over the solution developed.

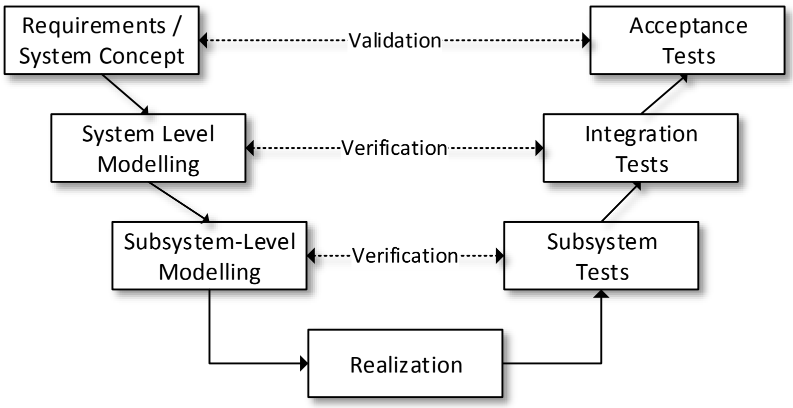


Figure 2.5: V-model using model-driven engineering techniques.

The approaches described above are process-oriented. However, it is also possible to adopt an artefact-oriented approach, where models are additions to existing process and are used as tools to facilitate collaboration. Such an approach is proposed in [93] and it uses SysML and UML to specify requirements for CPSs. This could be a perfectly valid way to fit some of the modelling activities proposed in this thesis within an engineering development process existing in an organization. In this work, we take a V-model lifecycle as the starting point in order to facilitate the explanation of the contributions in a well known and clearly defined systems engineering context.

2.6 Summary

This chapter has provided the necessary background to understand the core contributions of this thesis. The concept of CPSs has been introduced followed by an explanation of energy consumption and the presentation of energy saving techniques relevant to this work. Additionally modelling languages and methods for software and physical systems have been introduced followed by systems engineering lifecycles that make use of these techniques to construct CPSs. The following chapter takes as starting point the concepts presented here and explains how this thesis addresses their current limitations with regard to energy and power consumption analysis in the systematic development of CPSs.

3

A System-Level Methodology for the Energy-Aware Design of Cyber-Physical Systems

3.1 System-Level approach to the energy analysis of Cyber-Physical Systems

CPSs are composed of several kinds of subsystems, the most relevant being electromechanical subsystems, computational subsystems and communication subsystems. In order to get a holistic view of the energy consumption in the CPS under study all the subsystems shall be taken into consideration. Only by having this overall perspective it is possible to compare the energy consumption across the different subsystems that compose the CPS and determine where potential optimization efforts should be focused in order to lower the overall system's energy consumption. Additionally, these subsystems normally have interactions between them, implying that the way subsystem A functionality is realised might have an impact in subsystem B's energy consumption and performance. For example, imagine a CPS in which a mechanical plant is supervised by a controller through sensor and actuator units. These units are wirelessly connected to the hardware platform executing the controller software. This scenario is represented in Figure 3.1. In this relatively simple case it is not possible to design the control software without taking the characteristics of the wireless communication channel into consideration. In addition, the control software cannot be designed without considering the specifics of the mechanical plant. Finally, the mechanical plant cannot be designed without considering how the actuator and the sensor units are going to operate. On top of all these considerations one has to consider the energy consumption factor. For example, implementing a specific power saving policy on the control software might result in a delayed response to sensor readings and potentially result in a slower action via the actuators. This can have consequences on the energy consumption in the mechanical plant, and as a consequence consume more energy that was saved on the control software side.

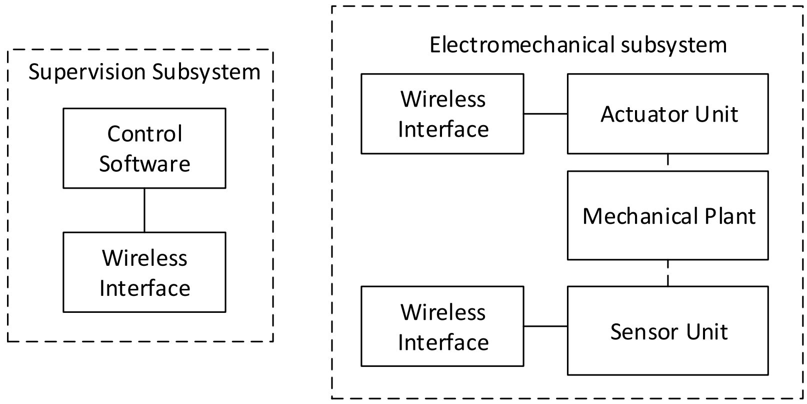


Figure 3.1: Example of a CPS featuring subsystems of different nature.

In order to analyse this kind of scenario, address the energy related issues as well as other challenges present in the design and development of such a system, the engineering disciplines involved should be capable of:

1. Achieving a holistic view on energy consumption. This kind of overall perspective on energy consumption allows comparison of systems of different natures developed by different disciplines within a single view.
2. Coping with the complexity associated with the design of this kind of systems. CPSs are composed of multiple subsystems that present their own design and implementation challenges not only from the energy consumption viewpoint but also from the functionality perspective. Being able to select the relevant aspects, focus and scope of the problem must be an integral part of the approach to tackle the design of CPS in which energy or power consumption is an issue.
3. Having a realistic yet manageable way of addressing the energy and power consumption in the different kinds of subsystems, so it can be analyzed systematically.
4. Having a way to gain confidence in the system design. This implies that the different disciplines representing the subsystems should be able to

validate and verify their design decisions, especially the ones regarding energy and power consumption.

5. Exploring the design space and therefore being able to justify why their design choices are the most appropriate.

In order to address capabilities 1 and 2 the application of abstract modelling is proposed. This is first conducted at the system level and later on at the subsystem level. This modelling is focused on capturing the functionality the system and subsystems are providing. In order to address the energy consumption in more detail and therefore address capability 3 a number of modelling approaches and techniques for the different kinds of subsystems are proposed. These address electromechanical subsystems, computation and communication subsystems respectively and are presented below. These models are developed using a heterogeneous approach. This means that several modelling notations are used depending on the engineering discipline conducting the modelling activities. This allows the engineers representing the different disciplines involved to use a notation which they feel comfortable using and that permits the creation of accurate models within their concrete application area. The challenge there is to reconcile a number of models developed using different modelling notations so they can be co-simulated, making it possible for the different disciplines involved in their creation to collaborate in an easier manner. As discussed in Section 2.3 there are co-simulation engines that can be used to address this issue. A different approach is to use languages that enable the creation of domain independent system representations, such as SysML. These kinds of languages cannot be used to produce models at the same degree of fidelity as domain specific ones and the kind of analysis that one can conduct is more focused around system architecture and hierarchy. However, they bring the advantage that they are relatively easy to learn by the different types of engineers involved in the CPS design and that these models can be used as a communication mechanism. Finally, and through the simulation of the models produced throughout this model-driven engineering approach it is possible to address capability 4 and produce the necessary evidence to support energy-related design claims.

The approach described above is framed within the model-driven engineering V lifecycle introduced in Section 2.5 and formulated as a number of ”add-ons”. Therefore, a contribution of this work is:

> **Contribution 1.** Additions to the model-based engineering V lifecycle so it is possible to address the energy consumption through a holistic heterogeneous modelling approach.

Figure 3.2 shows where in the V lifecycle the additions are located. These additions are:

1. **Energy consumption evaluation and interface definition:** This addition consist of the creation of a prioritized list of subsystems to be analyzed depending on their energy consumption. This enables focusing the optimization efforts where most energy is consumed. This prioritization is initially done through different traditional engineering processes (see Section 2.5) as well as by conducting experiments as defined later. Additionally, the interfaces among the different subsystems that compose the CPS are defined. This settles the basis for collaboration between the different disciplines involved in the development of a CPS. Any change on the interfaces would require the interaction of the disciplines involved in the subsystems under consideration.
2. **Experiments and energy-oriented modelling:** This addition consist of capturing how energy is spent at the overall CPS as well as at the subsystem level. In order to do so a number of specific modelling approaches and techniques are proposed later in the text. Additionally, experiments as defined below, are introduced, so the notion of energy and power consumption can be incorporated in a more accurate way into the models. The simulation of these models allows to study the performance of the system with regard to energy consumption and to analyze different candidate solutions for each subsystem. This enables the exploration of the design space in a cost effective way.
3. **Hardware In the Loop simulation:** HIL simulation is incorporated in order to facilitate that transition between models and subsystem realization. Additionally, HIL facilitates verification activities, conducted on the right hand side of the V lifecycle.
4. **Evidence production:** The application of this approach facilitates the production of information that can support the low energy consumption related claims. This can be coming from model/co-model/HIL simulation as well as from experimental measurements conducted over the final subsystem/system realizations.

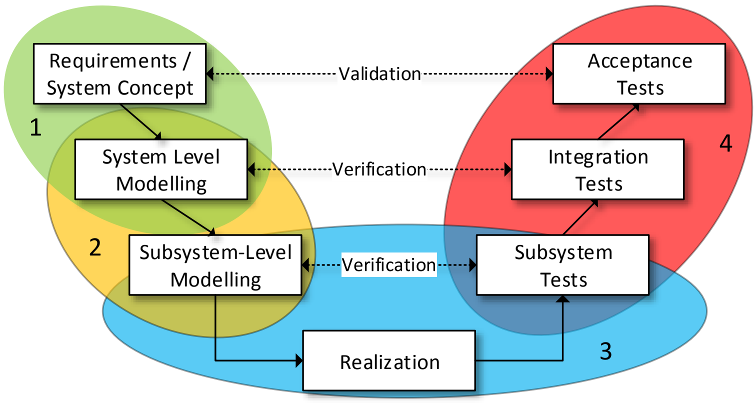


Figure 3.2: V lifecycle with the additions proposed here.

This approach to modelling and overall system understanding uses both theory and practice through experiments. These are classified in two different categories:

Explorative prototyping: in this kind of experiment part of a subsystem is prototyped by physically building it. The purpose is to study specific aspects that can later be incorporated into the model or alternatively to validate a given hypothesis regarding the operation of that part of the system. This can be related specifically to energy consumption or it can be used to study any other functional aspects of the system (that are ultimately relevant to energy consumption).

Measurement and profiling of existing elements: in this kind of experiment a model of the subsystem under study is created empirically rather than by developing a theoretical model analytically. Such an approach can be beneficial when having to use ready-made elements such as subsystems to be integrated or components to be used. Possible measurements that could be targets for such a study would be: variables with direct or indirect impact on the energy consumption, outputs provided by the system or time-related measurements among others.

The methodological approach resulting from the combination of a V model together with the additions proposed above facilitates the design and development of CPS with a limited energy budget. In principle, these CPS can be of any size, but this approach has only been tested in this work with low to medium complexity CPS running on batteries. The remainder of this chapter presents the different kinds of modelling approaches as well as tool developments produced in this thesis in order to address the different kinds of subsystems that compose CPSs.

3.2 Electromechanical energy analysis

Electromechanical systems are a key part in many CPSs and often responsible for delivering the functionality for which the system was initially conceived. Additionally, they are usually composed of an electromechanical structure and by some kind of controller. The way such controllers operate the electromechanics can have a significant impact on the total energy consumption. Also different electromechanical architectures, such as different component arrangements or the introduction of new kinds of sensors or actuators can facilitate energy savings.

Energy consumption in electromechanical subsystems is mostly among the highest in CPSs and therefore it is the main kind of subsystem to address in order to optimize it. The modelling approach tackling these kinds of systems is heterogeneous, since we use different modelling notations to represent different system aspects (which can be in the Continuous Time (CT) or Discrete Event (DE) domains).

Contribution 2. Guidelines for the representation of energy and power consumption in electromechanical subsystems using co-simulation.

The application of this methodology has been structured as a sequence of four main phases that are described below:

CT modelling: At this phase the electromechanical aspects of the system are studied. These are represented using a CT formalism. Preliminary CT models should capture the core physical process in the system and define what variables are going to be used by the control logic, to be studied from the DE side. Preliminary model validation and verification can be conducted within the CT simulator without requiring a complete representation of the control logic. Modelling until this point is focused purely on capturing the functionality and does not incorporate the notion of energy consumption.

DE modelling: At this phase the control logic responsible for the supervision and regulation of the electromechanics is represented. This logic is represented using a DE formalism. Preliminary DE models should perform basic interaction with the electromechanical subsystem such as begin able to command all the actuators and read all the sensors (i.e. being able to use all the monitored and controlled variables properly). Once this preliminary interaction between the models has been captured, it is possible to conduct more sophisticated analysis. As in the previous modelling stage, the models produced in here do not incorporate any notion of energy consumption and are purely focused on capturing the functionality.

Model instrumentation: At this phase the models are modified in such a way that incorporate the notion of energy consumption. Instrumentation in this case is primarily conducted in the CT models, given the fact that energy consumption in the DE side is typically negligible if compared with the one on the CT side. It is advised to instrument the CT models at the component level. In this way it is possible to evaluate the contribution at the system-level of each individual component. This allows the identification of potential optimizations in particular system components. If one of the design objectives is the creation of a power or energy-aware system the control logic can be made aware of the power and energy consumption in the CT-side through the appropriate monitored variables.

Model simulation and analysis: At this phase the CT and DE models form a co-model that can be executed in a co-simulation. Here the target is to explore different system configurations and monitor the system performance versus the power and energy consumption. Concrete design goals depend on the specifics of the system under study, but some of the aspects that could be considered would be: instantaneous power consumption under different scenarios, energy consumption per service offered by the system, calculation of coarse-grained power and energy budgets, determination of expected battery life-time, design of the system behaviour under low battery levels (such as controlled degraded system performance), design of energy saving techniques or design of power and energy-aware operation among others.

The model-based engineering approach becomes especially relevant when it comes to develop electromechanical subsystems. Thanks to modelling and simulation it is possible to study system architectures and configurations that otherwise would be very expensive and time consuming to prototype. Even though this phase is targeting specifically electromechanical subsystems, it presents an approach that could potentially be applied to other kinds of subsystems. For example, a co-modelling approach could be interesting if the focus would be studying the performance of an electronic unit represented in the CT domain, and its supervision from the software side, represented in the DE domain.

3.3 Computation/embedded software energy analysis

Whereas electromechanics presents the ”physical” part of a CPS the software and hardware components constitute the ”cyber” part of a CPS. The energy consumption in the computational part of the system also needs to be considered. This analysis is mainly focused on how and when to use the sleeping facilities provided by the hardware executing the system’s software and how that impacts the design of its logic. Since such phenomena is primarily a discrete aspect of the system it is best described using a DE formalism such as VDM-RT.

Contribution 3. Extension of the VDM-RT language for the representation of energy and power consumption in computational subsystems by enabling sleeping states.

Contribution 4. Guidelines for the representation of CPU low power states at an abstract level, that corresponds to the way real CPUs operate.

There are multiple ways to take energy consumption of computation into consideration. These vary in terms of procedures (experimental methods, simulation, profiling) as well as in abstraction level (transistor-level techniques, CPU micro-architecture, OS Level, application software). The approach that is proposed here is to study the application duty cycle aiming at sleeping the CPU core when there is no logic to be executed. It is also possible to study the possibility of using different execution frequencies for the CPU core and therefore enabling the study of frequency scaling. However with the current hardware trends it could be more beneficial to run the CPU at a high speed so it can finish its computations earlier and sleep its core as much as possible [19].

The starting point is the CPU execution environment provided by the VDM-RT environment. These CPUs do not feature any representation of the typical energy saving facilities that are normally present in today's modern microcontrollers as described in Section 2.2. The initial approach [P52] consisted of the explicit modelling of an additional scheduling policy running on top of the CPU preventing the pre-emption of a thread on a CPU core that was in a virtual sleeping state. This approach was formulated as a design pattern structure that could be used at the modelling level. Even though this is effective from the functional point of view it makes the modelling phase more complex. In order to address this limitation the VDM-RT language has been extended with a **sleep** instruction and the necessary extensions to the scheduler to support it [P50]. This implies that the VDM-RT scheduler does not pre-empt threads in a sleeping CPU until it is active again. In Listing 3.1, an example is shown in which this instruction invoked from a CPU called microcontroller unit (mcu). Additionally the model captures the time it takes to execute the application logic as well as the time it takes to wake up through **duration** statements.

```
duration (APP_TIME) ApplicationLogic();
System`mcu.sleep();
-- Blocks until activated externally
duration (POST_WAKE_UP) PostWULogic();
```

Listing 3.1: High-level representation of the embedded software sleeping the CPU.

Finally, the sleeping CPU has to be woken up from an entity external to the processing core through an invocation of the **active** instruction, see Listing 3.2.

```
System`mcu.active();
```

Listing 3.2: Static call waking up (activating) the `mcu`.

The entity responsible for waking up this core could be a sleep timer or an external interrupt. We have presented [P50] a modelling pattern to capture both strategies in VDM-RT. This pattern uses an architecture of two VDM-RT CPUs, one of them configured as a hardware block and a second one configured as a processing core running software. The hardware block is responsible for waking up the processing core from its sleeping state, which is determined by the application logic that its running.

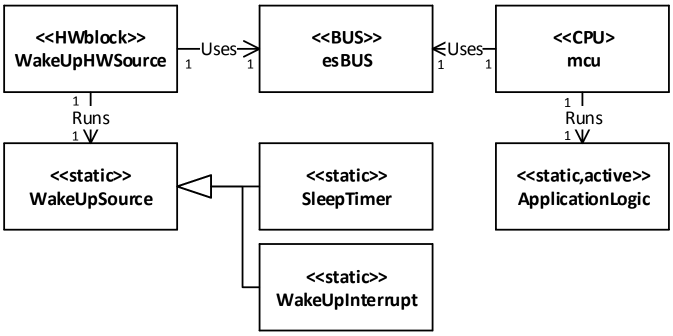


Figure 3.3: Modelling pattern for sleeping CPUs.

The models created with these facilities can be simulated over time and produce a simulation log where it is indicated when, for how long and under which mode the CPU has been sleeping. This, together with datasheet figures as well as previous measurements produce energy consumption estimations over a period of time.

From the methodological point of view, these modelling constructs shall be used in the following way:

1. **Determine the Real-Time requirements:** At this phase it is specified the real-time constraints the application must satisfy. Timing constrains might come from different sources such as: time estimations based on previous experiences in systems design, interfacing requirements imposed by external systems or timing constraints based on the ones present in similar products. Time requirements can be expressed in VDM-RT through the application of invariants. Additionally the modeller can benefit from the application of the constructs `duration` and `cycles` in order to represent the time penalties associated with the execution of different parts of the logic represented in the model.

2. **Determine the CPU clock frequency:** At this phase the approximate optimal operating frequency for the CPU that will run the control logic is determined. This can be done by modifying the clock frequency of the VDM-RT CPU and determining whether the real-time constraints

are violated or not through model simulation. Here the main issue is to decide between two possible strategies:

- Lower clock frequency, that will imply slower program execution and lower power consumption in the active state. This option could potentially require having the CPU active for a longer period of time in order to finish the computations required. The sleeping period therefore would be shorter.
- Higher clock frequency, that will imply faster program execution and higher power consumption in the active state. This option could potentially enable the CPU to finish their computational task sooner and therefore have a longer sleeping period.

3. **Determine the sleep strategy:** At this phase it is determined which out of the possibly multiple sleep modes the application is going to use. This selection might depend on the components that are needed during the CPU sleep modes as well as on the wake-up times. The wake-up delay might have impact on the real-time requirements that has to be taken into consideration. Additionally it might be necessary to revisit step 2 since this wake-up time might be dependent on the CPU frequency.

As in the case of electromechanical subsystems, the exploration of different configurations and system architectures is simplified if conducted at the simulation level. This facilitates a wider exploration of the design space and eliminates problems always present in experimental approaches such as noise and repeatability.

3.4 Communication energy analysis

As in the computation case, VDM-RT does not incorporate any notion of energy consumption with regard to communication. On the other hand, as explained above in Section 2.3, VDM-RT provides the necessary abstractions to represent distributed systems. Finally there were no clear indications regarding the representation of networks in VDM-RT, since the notion of BUSes and CPUs were initially targeting the representation of distributed embedded systems rather than general communication networks.

Contribution 5. Guidelines for the representation of energy consumption in communication using the distributed aspects of VDM-RT, as well as the representation of small-scale networks at an abstract level.

The representation of small scale networks in VDM-RT is built on top of the language by explicitly representing the connection between nodes by the application of visibility maps. These maps contain a separate entry for each node in the network and a list of the nodes that it has within range. This modelling pattern brings the possibility of representing different kinds of network topology and gives the possibility of experimenting, to some extent with protocols across different levels in the communication stack.

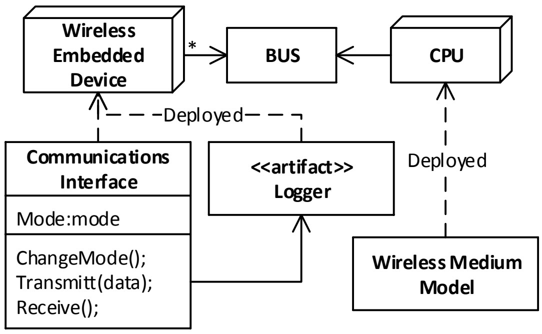


Figure 3.4: Deployment diagram showing a VDM-RT network model.

The approach to energy consumption analysis is focused on keeping track of the usage of the network interface in terms of data transmitted and received. This, in principle, does not require any modification to the modelling constructs provided by VDM-RT and it can be done by simply supervising the amount of data that has been put through a VDM-RT BUS. Additionally, it is possible to represent multiple operational states that feature different energy consumption by simply modelling those explicitly in VDM-RT. Finally a logging functionality can be easily incorporated in the models as it is done in the computational energy consumption analysis case.

This analysis could be applied to both wired and wireless networks (even though naturally the usage of wireless interfaces is more common when considering battery powered devices). This methodology targets small-scale networks (such as Body Area Networks) as well as the characterization of how a single device handles its communication within networks of any size.

The methodology proposed to design communication subsystems is composed of the following phases:

1. **Abstract protocol modelling:** At this phase the main focus resides on capturing the high-level interactions between the systems involved in the communication. Critical points to capture here are who is initiating the communication, what kind of information communication scheme is used (such as data push or pull) as well as time related requirements.

2. **Definition of the application-level messages:** At this phase the Protocol Data Units (PDU) that are going to be used during the communication are defined. Based on this it is possible to determine the approximate size of the packets that are going to be sent in the different communication scenarios defined previously.

3. **Message energy consumption profiling:** At this phase it is determined how much energy is spent when transmitting and, if relevant, receiving the different messages that are involved in the communication. This profiling can be conducted in different ways, such as: experimentally by prototyping the transmission and reception of information, re-using experimental models produced by other researchers or by using manufacturer specifications. The prototyping approach is suggested since it allows to gain a more thorough insight into the technologies used.

4. **Model instrumentation and simulation:** At this phase the models are instrumented so they incorporate the notion of energy consumption. This allows the generation of communication logs that not only show the interactions between the devices involved in the communication but also the energy consumption. This total energy consumption is calculated by adding the consumption associated to the transmission and reception of each individual message, profiled in the previous step.

As in the case of energy consumption in computation, analysis in communication can be carried out at different levels of abstraction. The intention behind this approach is to evaluate the impact of different protocols at the application level in energy consumption. For the analysis of energy consumption at lower levels in the communication stack, it is advisable to conduct simulations using more dedicated simulators, such as MiXiM [15] or OMNET++ [16] and incorporate to the extent possible the simulation results to the high level models presented in here. The latter also incorporate basic models for simulating energy consumption at the node level that can be useful in this context [29] .

3.5 Hardware In the Loop simulation

The approach proposed above relies heavily on the creation of models to conduct energy consumption analysis. However, these models have to be, eventually, transformed into a system realization. This process is not trivial and it is conducted both automatically as well as manually in both hardware and software fields. Additionally, it might be interesting to have the possibility of integrating part of the models with partial system implementations in order to conduct validation and verification as well as to gradually move models to realization.

The 20-sim platform for CT modelling incorporates advance HIL simulation facilities as well as the possibility of generating code based on the models created using the bond graphs formalism. On the DE modelling side and taking VDM-RT as notation there has been no previous work on creating a HIL setup that can be used to combine models with partial system implementations. Therefore, one of the contributions of this thesis is:

Contribution 6. Extension of the Overture platform to partly support Hardware In the Loop simulation, therefore facilitating the combination of models with partial system realizations.

The HIL setup created in this work had as main objectives: The possibility of conducting execution time measurements that could be incorporated into the VDM-RT model and the possibility of having a VDM-RT model interacting and/or controlling the execution of a certain logic running in hardware or software in an external Device Under Test (DUT). Such a functionality has been achieved by incorporating elements able to both monitor and coordinate simulation and execution in real targets. The hardware architecture of the HIL system developed in here is shown if figure 3.5. The system is composed of the following main blocks:

Workstation: Runs the VDM-RT models in the Overture environment as well as the necessary logic for the interaction with external devices. It also uses the necessary interfaces to manage the rest of the components involved in the simulation.

DUT: Device Under Test, is the external hardware that is in the loop with the simulation. This can feature both hardware and software components as well as software functionality to enable the management from the workstation side.

Stimuli provider: Provides external signals to the DUT in order to represent stimulus from external components or environmental conditions.

Logic Analyzer: Monitors the response of the DUT in its different logical outputs, it measures execution times by reading duration pin status and monitors other kinds of digital signal outputs such as PWM. It can be read from the workstation in order to evaluate the response of the DUT.

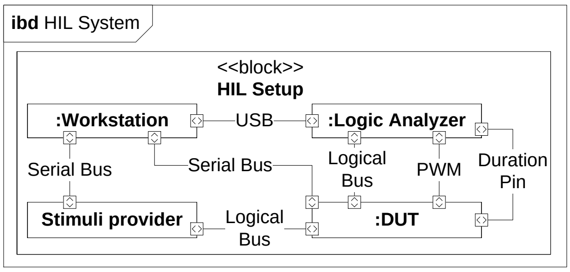


Figure 3.5: SysML internal block diagram showing the composition of the HIL setup.

Additionally, focusing on the software and taking into consideration the workstation side, it has been necessary to make it possible for the overture tool to interact with low-level hardware related aspects, such as communication with external Logic Analyzer and custom hardware components through APIs and handling of serial communications with other devices. The work conducted in the HIL area cannot be considered a finished product but it demonstrates that HIL can be in fact realized within the Overture context and that it can facilitate transition from models to realization, the increment of model fidelity as well as testing. This work will be extended in the INTO-CPS project[1] and further details on this will be provided in the Section 5.5.

[1] INTO-CPS official project website: `http://into-cps.au.dk/`

3.6 Summary

This chapter has presented a new methodology to address the development of CPSs in which energy consumption is a concern. The methodology is model based and uses prototyping combined with heterogeneous modelling in orther to address the subsystems present in a CPS. This methodology will be demonstrated in practice through its application in two case studies in the following chapter.

4

Case studies

4.1 Introduction

The methodology and modelling techniques presented in the previous chapter have been applied to two different case studies from the medical domain. Medical devices are typical examples of energy-constrained, battery powered CPS [95]. New technologies in the computation, communication and electromechanical fields have made a new generation of wearable medical devices possible [109]. Additionally, energy harvesting techniques have been considered as possible solutions to increase device autonomy [8]. From the methodological side, model based approaches for Design Space Exploration (DSE) and Verification & Validation (V&V) targeting specifically wearable solutions have been proposed [4, 88].

The case studies discussed here are: a compression stocking to treat leg venous insufficiency (e-Stocking) and a wearable heart monitor. These cases complement each other since the first one is more electromechanically intensive while the second one is more focused on sensing, communication and communication. The e-Stocking case study is also more complex and more extensive work has been conducted developing prototypes of different kinds.

4.2 e-Stocking: a compression stocking to treat leg-venous insufficiency

This case study is based on the European Ambient Assisted Living project e-Stocking, in which the aim is to create a compression stocking to treat leg-venous insufficiency[1]. This system is required to deliver a compression that ranges from 40 mmHg at the ankle level to 20 mmHg below the knee. This wearable healthcare device is composed of mechanical, software and communication subsystems. A prototype of the device can be seen in Figure 4.1.

[1] e-Stockings project official website is: `http://www.e-stockings.eu/`.

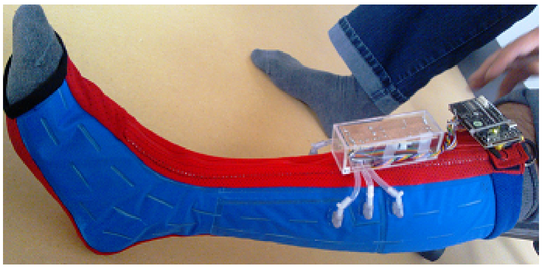


Figure 4.1: The e-Stockings prototype.

An overview of the different parts of this system can be found in [P57]. Additionally, this device incorporated a network interface and performed a number of communication operations, as presented in [P44].

This system presented a number of design challenges, where the most relevant ones in relation to this PhD project are:

- The operational time should be between 12 to 14 hours, enabling the patient a complete day of use without requiring a battery recharge.
- The pressure delivered to the limb shall be constant and within the prescribed range. This implies that the stocking shall feature a regulation mechanism that makes sure that the delivered compression is inside the specified interval.
- The stocking controller should be able to communicate with a smartphone acting as an internet gateway and/or a configuration tool.

The methodology presented in chapter 3 was applied to tackle these challenges both from the functional and energy consumption point of view [P56]. This constitutes the following contribution of this PhD:

Contribution 7. The application of the methodological additions to the design and implementation of a compression stocking to treat leg-venous insufficiency.

4.2.1 Application of the methodological additions

In Section 3.1 a number of methodological additions to the traditional V lifecycle were introduced. These additions have been applied in this case study as follows:

Component	Current draw	Voltage	Power consumption
Pump	110 mA	3 V	330 mW
Valve	120 mA	3 V	360 mW
Manometers	1.4 mA	3.3 V	4.62 mW
CPU Active / Sleeping / Hibernating	20 mA / 10 uA / 10 nA	3.3 V 3.3V	66 mW / 33 uW / 33 nW
Radio Tx/Rx	35/40 mA	3.3 V	115.5/132 mW

Table 4.1: Overview of the typical components power consumption.

1. **System-Level Concept and System-Level Modelling:** In this phase the system was represented from a top-level perspective and the different subsystems were identified. Additionally the interfaces between the subsystems were defined and preliminary inspection of the power consumption figures of the different components was conducted. This implied the application of the methodological additions 1 (Energy consumption evaluation and interface definition) and 2 (Experiments and modelling). As a result of this it was possible to get an overview of different power consumption figures as presented in Table 4.1.

2. **Subsystem-Level Modelling and Simulation:** During this phase the different subsystems that comprise the compression stocking were studied separately. Different experiments and profiling were conducted in order to have more accurate figures and better understanding of power consumption. Additionally the application of modelling and simulation helped to explore the design space and study different candidate architectures and/or configurations for each of the subsystems that compose the compression stocking.

3. **Realization and Evidence Production:** During this phase the selected designs in the previous phase were realized and different, final verification measurements were conducted. These helped in supporting the predictions provided by the models as well as act as additional piece of evidence to support quality related claims and/or the correct functionality implementation. This is part of the methodological addition 4 (Evidence production).

4.2.2 Electromechanical modelling

The mechanical side of the compression stocking has been represented in a 20-sim CT model. The plant control logic has been modelled in VDM-RT. The combination of the two models has been co-simulated using Crescendo. An overview of the mechanical side is given in Figure 4.2 and its components are described below.

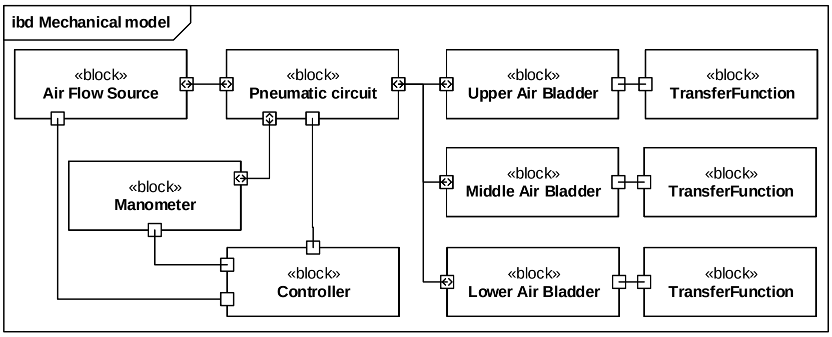


Figure 4.2: SysML Internal Block Diagram of the mechanical model.

Air Flow Source: represents the pump that generates the air flow to be directed to the air-bladders to build pressure. It is controlled by the software controller and the effective airflow can be modulated.

Pneumatic Circuit: represents a concrete arrangement of valves and tubing connections. The valves are controllable from the software and this makes it possible to direct the air-flow to a single air bladder. The valves are also responsible for the venting of the air bladders, however this cannot be modulated.

Manometer: is an air pressure sensor that can monitor the level of compression in the different air bladders. Due to the way it is connected through the pneumatic circuit a single manometer can be used to monitor any of the three chambers.

Air Bladders: represent the chambers integrated in the stocking in which pressure builds-up. Their air intake can be locked without having to energize the valves.

Transfer Functions: are different mathematical expressions that determine how the pressure in the air chambers map to the pressure over the skin, representing the effective compression. They can be replaced depending on the compression principle under evaluation and the rest of the model still can be reused.

Controller: contains the necessary interface definitions to communicate the mechanical model with the software control logic modelled in VDM-RT.

Figure 4.3 presents the top level representation of the 20-sim plant model. The complete model connects two more bladder subsystems (shown with the dashed box) to the distribution valve but they have been removed in this case for clarity. The pneumatic circuit presented above is decomposed in a *Distribution valve* and one *Pass valve* per bladder subsystem. The *Distribution valve* is responsible for directing the air flow to the air bladder that has to be inflated. The *Pass valve* is responsible for locking the air bladder once inflation has been completed. Hence, in order to inflate an air bladder two valves have to be energized. The block *LegSegment* introduced the transfer function that maps the pressure built in the *AirBladders* with the pressure exerted over the leg. This model incorporates the notion of power consumption in the most power demanding components: the *Distribution valve*, the *Pass valve* and the *Pump*. When the models are simulated the power consumption figures are integrated over time, resulting in the energy consumption for each individual component. The models are instrumented so both power and energy consumption are monitored variables in the simulation but without having an impact on the simulation performance.

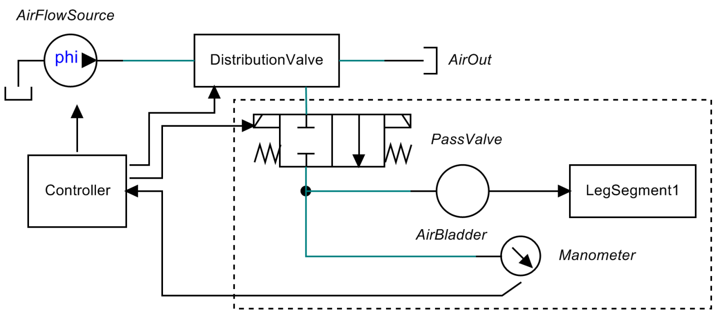


Figure 4.3: 20-sim CT model of the compression stocking.

The software controller modelled in VDM-RT contains the necessary interfaces to control the simulated sensors (manometer) and actuators (valve and pump). Additionally it features different regulation algorithms and configurations. These regulation algorithms aim at maintaining the pressure in the air bladders constant and constitute the core logic of the DE models. The most relevant ones are:

Simple regulation: that is based on conditional logical statements and do not modulate the pump air flow. This implies that the pump will be engaged fully in case additional compression is needed and the air bladders will be vented in case over-pressure occurs.

PID regulation: Proportional Integral Derivative [112, 39], that modulates the pump airflow depending on the deviation from the target pressure. Due to this behaviour it is said to be proportional in the inflation. Due to the mechanical construction of the system it is not possible to modulate the venting process and therefore it cannot be made proportional in the deflation. Therefore, a full PID controller cannot be incorporated in this system.

Listing 4.1 shows part of the model for a proportional regulation of the pressure in one of the air bladders. In this case the pump is driven proportionally to the error (`err`), which is the difference between current and target pressure (`setPoint`). Prior to engage the pump at specific rate, the controller sets the air distribution valve and opens the pass valve for the target air bladder (AB1).

```
err := setPoint - manometerAB1.getPressure();
if (err > threshold) then (
  pTerm = err * pGain;
  controller.airDistribution.airToAB1();
  controller.passValveAB1.open();
  controller.pump.setPumpRate(pTerm); );
```

Listing 4.1: Proportional regulation applied to air bladder 1.

4.2.3 Computation modelling

The control regulation logic introduced in the previous section is deployed as a software component, executed by the CPU integrated on the e-Stocking microcontroller. This CPU features a number of low power operational states to choose between when developing applications. Depending on the low power state used by the developer (Sleeping or Hibernating[2]) different kinds of wake-up mechanisms are available. In this work we have considered the following and most common ones:

Wake-up on sleep timer expiration: The CPU remains in a low power state until an internal timer overflows, generating an internal (within the chip) interrupt that wakes up the CPU. Applying this wake-up mechanism typically implies using the more energy demanding low power states such as the Sleep mode.

Wake-up on external event: The CPU remains in a low power state until an external event generates an interrupt that activates it. Applying this wake-up mechanism allow the use of Sleep or Hibernation modes.

These mechanisms facilitate the implementation of two different software regulation strategies for the e-Stocking case study:

Periodic regulation: in which the regulation logic is executed as a periodic thread. The period that determines how often the logic has to be executed can be determined by the study of the control requirements carried out during the mechanical modelling, presented in the section above. While the regulation is not executing, the processor can be put to sleep. This approach makes use of a *Wake-up on sleep timer expiration* strategy.

Event-triggered regulation: in which the regulation logic is executed once a pressure loss has been detected by smart sensors. The CPU can be put to sleep for an undefined period of time and remain in that state until it is notified by any of the sensors. This approach makes use of a *Wake-up on external event* strategy.

The energy consumption will depend directly on the strategy adopted and how it uses the low power features based on sleeping modes. These regulation strategies are implemented through two different system architectures, shown in the UML deployment diagram presented in Figure 4.5. These architectures are:

[2] Modern CPUs incorporate several low power modes, the most common being: a *Sleep* mode with a current draw within the order of microamps and a *Hibernation* mode, with a current draw of nanoamps and less reactive than the first one. Mode names vary depending on the manufacturer.

Architecture A: implements a periodic regulation strategy in which the CPU executes regulation logic and moves to sleeping state. After the sleep timer has expired it wakes up the CPU and executes the regulation process again. The system uses passive pressure sensors that have to be actively polled by the CPU.

Architecture B: implements an aperiodic regulation strategy in which the CPU hibernates until an abnormal pressure level is detected by the pressure sensors. This architecture makes use of Smart Pressure sensors, capable of monitoring the pressure levels independently of the main CPU. Once a pressure deviation has been detected by these pressure sensors they generate an external interrupt that wakes up the main CPU, that will finally execute the regulation logic. The power consumption of these sensors is higher than the passive sensors but still negligible and orders of magnitude below the power consumption of the CPU executing the control logic.

Common to both architectures are the `CPU` core and the `PWMDriver` block, that is responsible for generating the airflow to inflate the chambers.

Both architectures have been modelled in VDM-RT as shown in the deployment diagram presented in Figure 4.4. In this diagram we use nodes to represent the VDM-RT CPUs (stereotyped as <<CPU>>). In these CPUs we deploy different parts of the model to run independently and they are connected through VDM-RT BUSes. This model can be configured in two different ways to represent either Architecture A or B, by using the model of the `SleepTimer` or the `WakeUpInterrupt` respectively. The `Controller` class deployed in the node `mcu` models the logic that implements the regulation functionality presented in the previous Listing 4.1 as well as the logic that determines whether the CPU is sleeping or not.

4.2.4 Communication modelling

The e-Stocking case presents several communication scenarios [P44]. We have focused on the most relevant from the energy consumption point of view:

Health monitoring: In this scenario the stocking transmits current and historical data regarding treatment adherence and condition evolution.

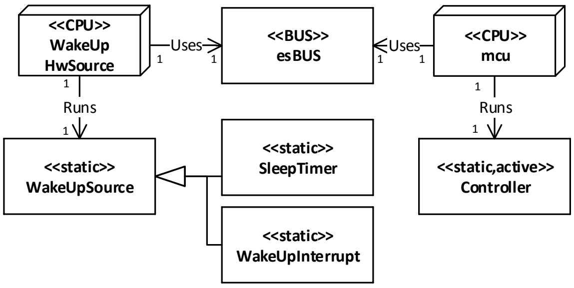


Figure 4.4: Class diagram showing the models general structure.

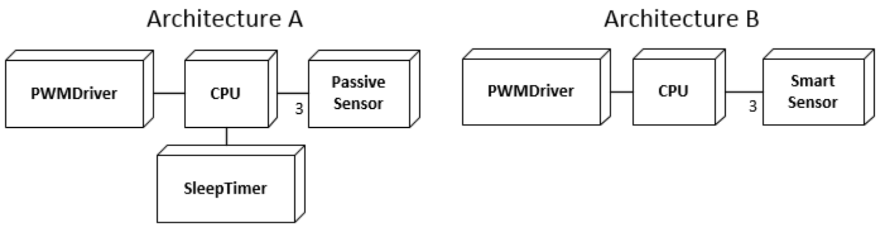


Figure 4.5: Deployment diagram with architectures for the two different regulation strategies.

Calibration and configuration: In this scenario the stocking settings that determine how the treatment is conducted (mainly pressure levels) are set through an external device.

In order to model these scenarios, we proposed the application of the Distributed Real-Time features of VDM-RT. The models developed using VDM-RT follow the structure presented in [P55]. This structure supports the modelling of small-scale networks such as Body Area Networks with several nodes interacting. In this case this structure can be simplified since there are only two nodes communicating. This structure is applied to this case as shown in the UML deployment diagram presented in Figure 4.6. The resulting model uses VDM-RT CPUs as execution environments in which the communication logic is run. The nodes represent the e-Stocking and an external client that connects to it in order to perform Health monitoring or Calibration and configuration as described above. The connection between the two nodes is represented through a VDM-RT BUS representing a 802.15.4 link. Each node is able to transmit data to the other by pushing it through the bus to the target receive buffer (`eSBuffer` and `clientBuffer`). Each node is able to receive data by reading its local receive buffers.

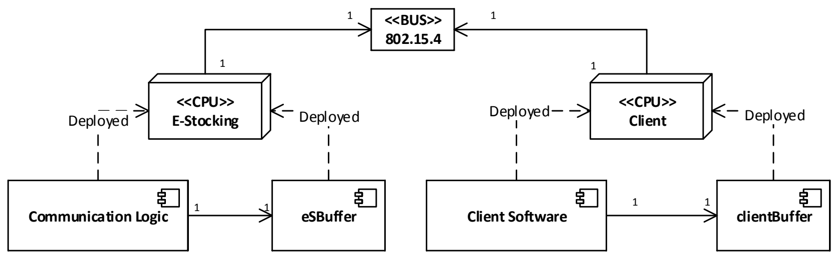


Figure 4.6: UML deployment diagram showing the structure of the VDM-RT communicaiton models.

4.2.5 Application results

The execution of the mechanical models has enalbed the evalutaion of different regulation configurations to determine their effectiveness as well as to compare them regarding their energy performance. We evaluated two different scenarios: compression from an idle state and regulation after an underpressure event occurred. Such an event might happen due to small leaks that have an impact on the pressure level during treatment. The simulation of the models allowed us to draw the following conclusions:

- The energy consumption during inflation and regulation is proportional to the time this process has taken and this is due to the particular configuration of the pneumatic circuit.
- This time can be reduced by inflating the air bladders as fast as possible, by configuring the PID controller with a high proportional constant,
- This implies that the valves that have to be triggered during inflation will be energized as briefly as possible, hence decreasing the energy consumption.

Applying this principle to the design of the software controller has lead to a decrement of ≈33% in energy consumption. Figure 4.7 presents the

Control Algorithm	System Energy Consumption [Joules]	Time Active [Seconds]	Pump Energy Consumption [Joules]
(a) p gain 2	323	222	103
(b) p gain 1	367	267	102
(c) p gain 0.5	472	372	102
(d) dummy control	348	232	116

Table 4.2: Energy consumed under different control algorithms.

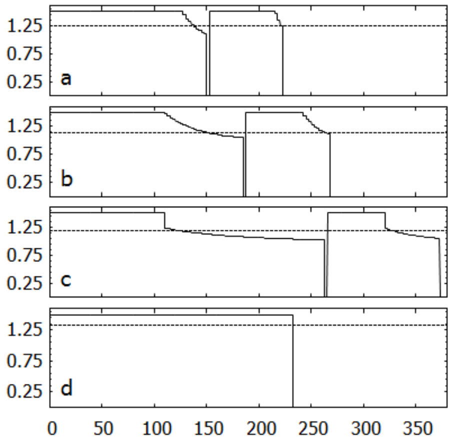


Figure 4.7: Power consumption of different control algorithms. y axis: power in watts, x axis: time in seconds.

evolution of power consumption over time and Table 4.2 the different total energy consumption estimations for each controller.

On the computational side, both architectures presented in Section 4.2.3 were simulated under a common scenario in which the regulation over a period of time of 200 ms was considered. In this scenario a pressure loss taking place at 150 ms was simulated. Both system models were simulated and their power consumption represented over time, producing the activity graphs shown in Figure 4.8. Based on these CPU activity graphs, taking into consideration CPU manufacturer specifications and basic CPU current draw measurements we were able to predict concrete average power and energy consumption figures.

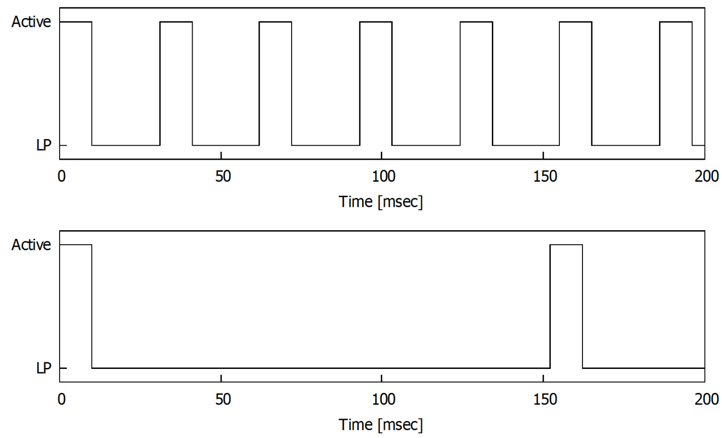


Figure 4.8: Activity over time predicted by the models in Architectures A (upper graph) and B (lower graph) in the scenario under study.

Finally, both architectures and control strategies were implemented on an ARM Cortex M3 based microcontroller and the energy consumption of both realizations measured. This resulted on the power consumption measurements shown on Figure 4.9 that allowed to validate the predictions showing that these exhibited an accuracy of 5% with regard to energy consumption.

On the communication side, several models capturing the communication process were created. The first two models captured the interaction between the devices (Model 1) and the information exchange (Model 2). Simulation results regarding energy consumption are produced later, after the notion of energy consumption has been incorporated through energy consumption message profiling and model instrumentation and simulation, as described in section 3.4.

Figure 4.10 shows the power consumption over time measured at the communication interface when transmitting and receiving a single byte. When integrated over time, this yields an energy consumption of 64 μJoules in transmission and 22 μJoules in reception. Due to message overhead and protocol specifics, there is practically no difference between transmitting messages that contains a payload between 1 to 10 bytes long. After conducting the profiling of the network interface presented here and measuring the time it takes to send and receive each message, Model 3 was created, featuring an estimation of the energy consumed on the communication side of the system. Basic predictions are shown in Table 4.3.

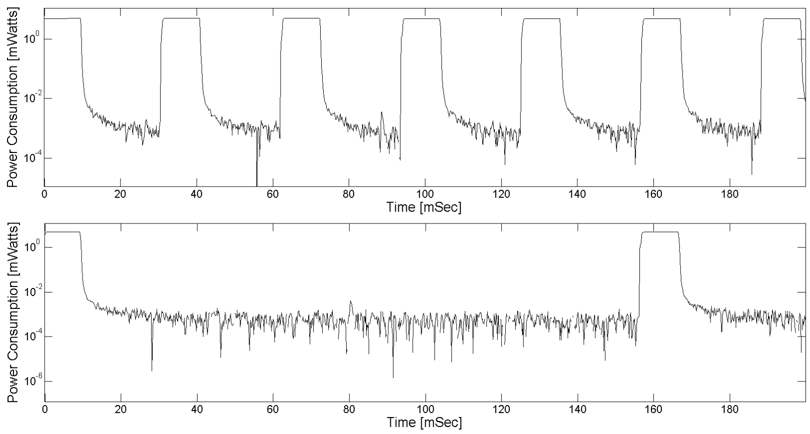


Figure 4.9: Actual power consumption of the control logic execution over time in Architectures A (upper graph) and B (lower graph). Note the logarithmic vertical scale.

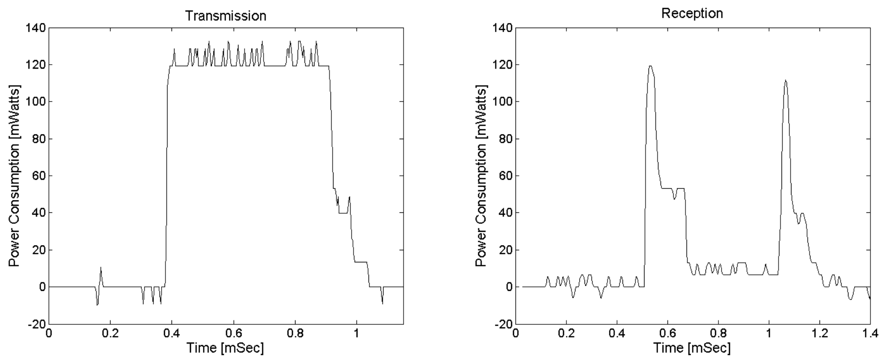


Figure 4.10: Power consumption for a single byte transmission and reception in the communication interface.

4.3 Wearable Heart Monitor

The second case study used in this thesis is based on the modelling, design and realization of a Wearable Heart monitor. There is extensive previous work on wearable ElectroCardioGraph (ECG) and pulse monitors both for medical, sports and research applications [3]. The novelty here resides in the energy analysis conducted through the application of the methodological additions

Model 1 **Interactions**	**Model 2** **Information**	**Model 3** **Time, Energy Consumption**
<– – RX <ConnectionRequest>	s0e	0.5 mSec +22 μJoules
– – >TX <ACK >	s1e	1 mSec +64 μJoules
<– – RX <SetPressures >	s53203040e	1.5 mSec +22 μJoules
– – >TX <ACK >	s1e	2 mSec +64 μJoules
<– – RX <Disconnection>	s2e	2.5 mSec +22 μJoules
Total communication time 2.5 mSec, Total energy consumption: 194 μJoules		

Table 4.3: e-Stocking communication model execution results for the configuration scenario.

presented in Section 3.1 rather than on the device itself. This led to the last contribution of this PhD thesis:

> **Contribution 8.** The application of the methodological additions to the design and prototyping of a wearable heart monitor.

The heart activity is monitored typically by studying the ECG signal, that represents the electrical activity of different parts of the heart over time. The ECG waveform is decomposed in several parts being one of them the QRS complex. Figure 4.11 shows three different QRS complexes corresponding to three different heartbeats. The HRV is defined as the separation in time between consecutive R peaks (RR distance) in the ECG and can be used to diagnose a number of medical conditions associated to the Autonomous Nervous System. The Instantaneous Heart Rate measured in BPM is defined as 6000 divided by the separation in time between two consecutive R peaks measured in milliseconds.

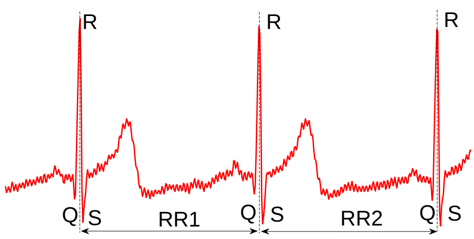


Figure 4.11: ECG waveform highlighting its main components.

Several application scenarios have been considered in order to be able to apply the modelling approach in more cases. The scenarios are:

ECG monitoring: In this scenario the monitoring device continuously sends the ECG waveform to a second external analysis/display device.

Heart Rate Variability (HRV) monitoring: In this scenario the monitoring device determines the HRV over a certain period of time. This can be relayed through the communication interface or stored in memory.

Pulse monitoring: In this scenario the monitoring device calculates the Beats Per Minute (BPM) and stores them in memory. This can be relayed through the communication interface or stored in memory.

The case study present a number of requirements that have an impact on the real-time operation of the system. These are:

1. The ECG should be sampled at 250 Hz.
2. The HRV study based on analysis in the frequency domain require a sampling frequency of 100 Hz[3].
3. The R peaks detection for HRV study in the time domain should be carried out with a precision of 5 ms.
4. The HRV study is conducted over periods of 2 minutes.

4.3.1 Application of the methodological additions

In Section 3.1 a number of methodological additions to the traditional V life-cycle were introduced. These additions have been applied in this case study as follows:

1. System-Level Concept and System-Level Modelling: In this phase the system was represented at a high level of abstraction and its main constituent blocks were identified. In this phase the specifications of the components were reviewed and a high level inspection of the different application scenarios was conducted. These steps, together with basic measurements (application of the methodological addition 2) led to a basic indication of where the main energy analysis should be conducted (application of methodological addition 1).

[3] Based on this relatively low sampling frequency the signal can be interpolated to increase its resolution later on, enabling a precise analysis of the HRV.

2. Subsystem-Level Modelling and Simulation: In this phase the different subsystems were modelled, paying special attention to the computation and the communication subsystems. Furthermore experiments were conducted to determine more precise energy consumption figures for the different functionalities to be implemented in the system and their results incorporated into the models (application of methodological addition 2).

3. Realization and Evidence Production: In this phase a partial realization of the system was conducted in order to validate the main model predictions (application of methodological addition 4). A complete implementation of the system was not produced since the main purpose of this case study is to apply the methodological additions and not to produce a full working prototype of the heart monitor.

4.3.2 System-Level modelling

At this initial modelling stage the system was analyzed from a top level and the key subsystems identified. These are presented in Figure 4.12 through a Block Definition Diagram (BDD).

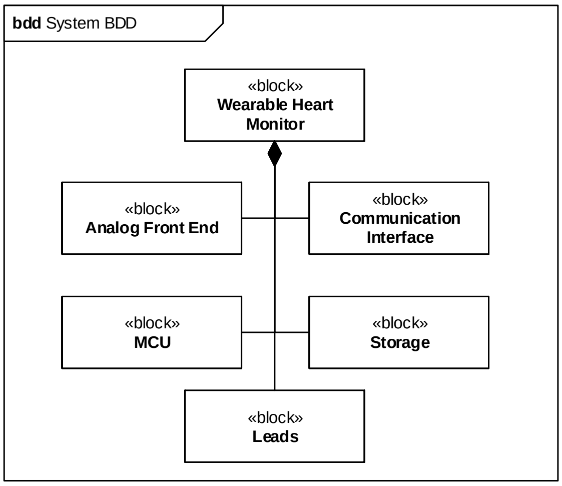


Figure 4.12: BDD of the ECG system.

Additionally a basic inspection of the power consumption of different system components under different operational modes was carried out.

These figures suggest that the power consumption figures can be similar for computation and communication subsystems depending on the chosen configuration and that it is not possible to determine directly which subsystem is going to be consuming the most. Additionally, the analog front-end presents a constant power consumption when acquiring the ECG signal and no energy saving techniques can be applied here (besides power gating it when not in use).

4.3.3 Computation modelling

Figure 4.13 shows a the UML class diagram of the VDM-RT model of the ECG software with the following components.

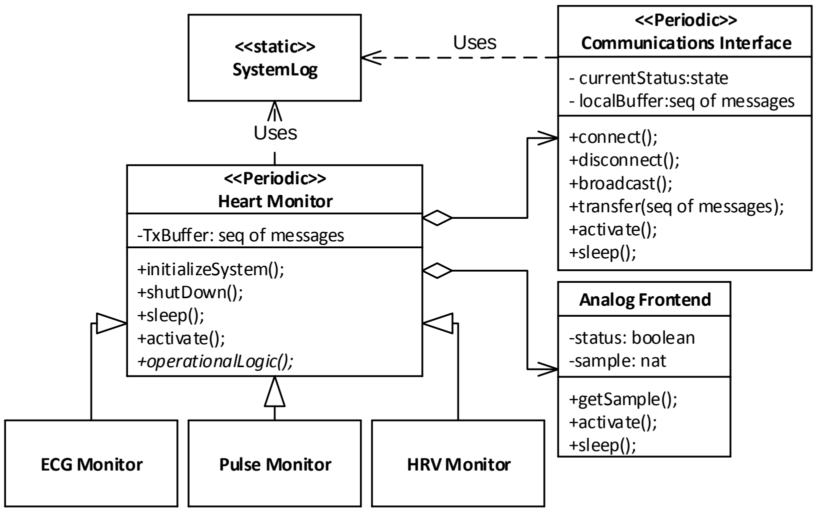


Figure 4.13: UML class diagram of the VDM-RT software models.

Heart Monitor: models the software logic that implements the core functionality of the device. This class is subclassed in three different behaviours depending on the functionality the device has to offer (ECG Monitor, Pulse Monitor or HRV Monitor).

Communication Interface: represents the subsystem responsible for data transmission to a second device, responsible for data presentation to the final user.

Analog Front-end: represents the subsystem that acquires, filters and process the heart signals. It provides as output an analog voltage that can be sampled and acquired from the Heart Monitor logic.

The modelling of the system from the computation point of view started with a high-level description of the software logic and the interfaces between the software system and the external hardware, which defined the model boundaries. Following this, abstract representations of the sequences of actions for each scenario discussed above was created, paying special attention to the time constraints foreseen for each particular scenario and the instructions in the logic that could potentially have a negative impact on those. The reason behind this focus is two-fold: first because real-time deadlines have an impact the system implementation and second because depending on the real-time performance required some energy-saving optimizations are not possible.

The ECG monitoring behaviour has been modelled as a periodic thread with an execution frequency specified by `ECG_MONITOR_PERIOD` (see Listing 2. Given the fact that the goal is to reach a sampling frequency of 250 Hz, the thread period has been set to 4 ms. The operations conducting the sampling and transmission or storage of a sample have to be completed within that period of time so the system can process the next sample on time (and therefore effectively provide the 250 samples per second). This implies that the duration of each of the operations executed during the sampling cycle has to be taken into consideration in the VDM-RT model.

```
thread periodic(ECG_MONITOR_PERIOD,0,0,0)(logic);
```

Listing 4.2: Declaration of the periodic real-time thread in VDM-RT.

```
public logic : () ==> () logic() == (
duration (GET_SAMPLE_DURATION) (
  sample := analogFrontEnd.getSample(););

duration(BUFFER_STORE_DURATION)
(txBuffer := txBuffer ^
[mk_CommInterface`message(time,sample)]);
```

```
duration(COMPARISON_DURATION) ();

if len txBuffer >= CommInterface`PACKET_SIZE then (
  commInt.transferData(txBuffer);
  commInt.dataReady();
  txBuffer := [];
  duration (TX_MESSAGE_DURATION) (skip););););
```

Listing 4.3: Logic executed within the periodic thread.

The operations have been prototyped and profiled in the concrete CPU (an Atmega 328 microcontroller). The kind of operations used in this sampling cycle and the actions they perform makes the scaling of the CPU processing frequency insignificant. The bottleneck in the sampling process is the data transmission through the serial interface, therefore it has been set to a maximum baudrate (115200) and therefore a frequency of 4MHz has been set, suitable for that particular baudrate. No further reduction in execution frequency is possible and therefore no further reduction of energy consumption through frequency scaling.

This basic model and analysis of the computation system time and energy performance, could be used as a base to explore other kinds of functionality such as the applicability of compression schemes over the samples and how these could be carried out without altering the real-time performance. Additionally it could be interesting to see if other actions could be performed incrementally during the available processing time between samples or if it could be feasible to skip a certain number of samples with a given period in order to carry out other operations.

The behaviour for the HRV and pulse monitoring case have been modelled in a similar way and profiled for the chosen CPU accordingly. In the HRV monitoring case two kinds of analysis are possible: 1) to store the ECG waveform in an SD card and perform an off-line analysis or 2) to conduct an on-line analysis by registering the difference between the arrival of the peaks in the ECG signal (R peaks).

The design for an on-line HRV analysis is based on having the analog front-end triggering an external interrupt every time a peak is detected in the ECG. This triggers an Interrupt Service Routine (ISR) that stores a time mark in a local buffer and puts the processor in an idle loop waiting for the next peak to trigger the interrupt again and repeat the process. The process of detecting the peak and triggering the ISR is very fast and conducted in less than one ms, therefore it does not compromise the RT performance of the system.

A possible approach to energy consumption reduction in this case is to have the CPU core sleeping as it was carried out in the previous case study. In this case the Interrupt triggered by the analog front-end could wake-up the CPU that would be waiting in a sleeping model. A review of the microcontroller specifications has shown that the wake up time of the CPU is close to 100ms rendering this energy saving approach unfeasible without compromising the real-time performance of the system. A possible alternative would be to use a CPU with a faster wake-up time, which could allow sleeping cycles between the heartbeats.

The pulse monitor has been modelled and implemented using this technique, allowing the reduction of energy consumption during operation yet providing correct system operation.

4.3.4 Communication modelling

The analysis of the computation subsystem shows that data is produced at different rates depending on the considered scenario. These figures are shown in Table 4.4. As it can be see there, the most energy demanding scenario is ECG monitoring, since it is the one with the highest data transmission requirements. It is critical that the data is sent through the communication interface as fast as possible so the real-time performance is not affected and no samples are lost due to buffer overflows. Additionally the device still has to operate in an energy-efficient way.

Operational Mode	Sampling Rate	Data Production Rate
ECG monitoring	250 Hz	500 bytes/sec
HRV ECG-based	100 Hz	200 bytes/sec
HRV time-based	2 Samples/sec	4 bytes/sec
Pulse monitoring	2 Samples/sec	4 bytes/sec

Table 4.4: Data production rates.

From the VDM-RT perspective the communication interface has been modelled as a hardware block that receives data to be transmitted. This means that the logic within the interface runs in a separate VDM-RT CPU configured with a processing speed at least one order of magnitude higher than the VDM-RT CPU running software. The logic of the communication interface is run periodically and can be configured to send a certain number of samples per packet (see PACKET_SIZE).

```
public COMM_INTERFACE_THREAD_PERIOD : nat = 1e5;
public PACKET_TX_TIME : nat = 4e5;
public PACKET_SIZE : nat = 20;
```

Listing 4.4: Configurable parameters of the VDM-RT model of the Communications Interface.

In order to minimize the overhead in the transmission of the data due to the construction of frames, it has been decided to send the information in packets of 20 Bytes (10 samples per packet).

Based on this packet size and the specification of the different communication interfaces considered as well as the protocols they implement, one could calculate the energy consumed for each packet transmitted. However (and using the experiment-based methodological addition presented previously) different communication interfaces have been profiled for the specific scenarios discussed in the models. These interfaces have been profiled for a data packet of 20 Bytes and its power consumption measured over time. Additionally, these measurements allow to produce more accurate estimates of energy consumption per amount of information transmitted, that can be incorporated into the models in order to produce precise energy consumption predictions for each usage scenario and communication technology. Figure 4.14 shows the power consumption over time measured at the communication interface for a packet of 20 Bytes. The numerical integration of this yields the energy consumption, which is shown in Table 4.5. These energy consumption measurements are incorporated into the model, which can be run with different data production rates (by modifying the sampling frequency) and therefore producing different energy consumption estimates for different configurations. These measurements have been conducted over the communications interface when transmitting the final packet, therefore including the overhead associated to error correction codes and protocol specifics. Even though these details are not captured explicitly in the VDM-RT model they can be taken into consideration when calculating the final energy consumption after the measured energy consumption figures have been incorporated into the models.

In this case the communication interfaces energy consumption has been measured directly and incorporated in the models, however one could use

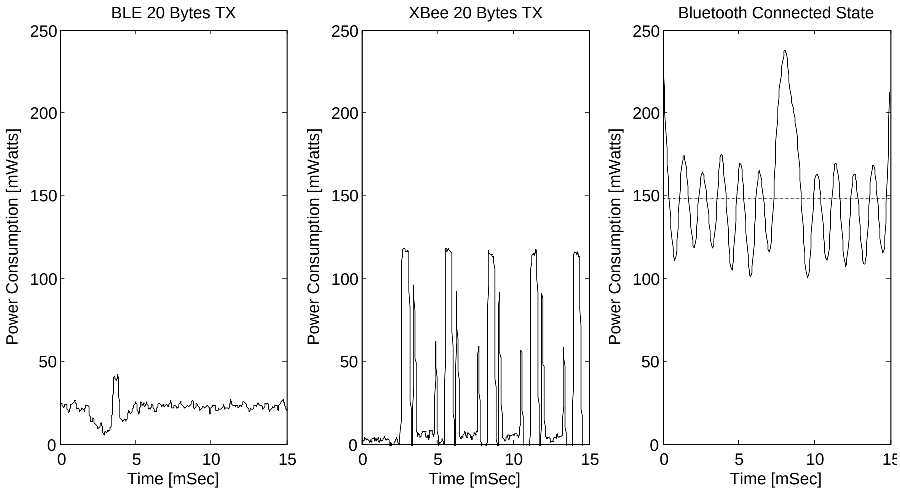


Figure 4.14: Communication interfaces power consumption over time.

Interface type	**Energy Consumption 20 Bytes Tx**	**Energy Consumption per Byte TX**
BLE Nordic Semic.	23 μJoules	1.1 μJoules
Xbee 802.15.4	390 μJoules	19.5 μJoules
Bluetooth RN-42	2220 μJoules	111 μJoules

Table 4.5: Energy consumption for each kind of communication interface considered.

generic energy consumption models provided by manufacturers [59] or reuse more complete experimental models created by other researchers [25].

Based on the simulation of the models and the energy consumption estimates it can be concluded that:

- Bluetooth presents a high energy consumption since it has to maintain the communication channel open continuously. Additionally the long connection times makes it impossible to perform periodic connection and disconnection. On the positive side bluetooth presents a very high data rate of several Mbits/sec, which could be useful in case data production rate would be higher.

- XBee is more energy efficient than bluetooth and allows to send small packets of data. The overhead in the packet is reduced yet it has to be taken into consideration when sizing the payload. From an application perspective this kind of network technology is not widespread and its interaction with other external devices is more limited, however previous research has presented gateways between this technology and smartphones [78].

- BLE is the most energy efficient solution in this case and offers a higher data throughput if compared with the XBee solution. It also offers a configurable packet size and a reduced header. This results in less data to be transmitted and a higher effective datarate. As shown in the model simulation is the most suitable communication strategy from the energy consumption point of view.

4.3.5 Application results

Based on the simulation of the models it has been possible to evaluate different system configurations and produce energy consumption estimates for the subsystems responsible for computation and communication in the ECG. In addition to those the Analog Front-end is also consuming energy while acquiring the heart signal but, since it presents a constant energy consumption no further analysis has been conducted. Based on the figures produced for each scenario it is possible to create energy consumption estimates for the whole system. For example, table 4.6 shows the energy consumption for each subsystem as well as the energy consumption for the whole system.

Subsystem	**Calculation**	**Energy Consumption per hr of ECG**
Analog Front-end, Active mode	3 mA $\times$ 3.3 Volts = 9.9 mW	35.64 J
CPU 4 MHz, Active mode	2.4 mA $\times$ 3.3 Volts = 7.9 mW	28.51 J
BLE UART mode	2.3 μJoules/Sample at 250Hz = 575 μJoules / sec	2.07 J
TOTAL energy consumption for 1 hr of ECG = 66.22 J		

Table 4.6: Energy consumption for the ECG monitoring application scenario.

Based on the energy consumption demands presented by the system it is possible to evaluate different battery technologies and configurations and determine which one is the most suitable for the application under consideration. In this case, and taking as an example the ECG monitoring case presented above one could calculate for how long the device could be powered with different kinds of batteries (see table 4.7). Here the choice between battery technology is not always the one that provides the highest battery lifetime, there might be other criteria for selection such as durability, weight or safety of operation however, these considerations are out of the scope of this thesis.

Battery type	**Avg. Voltage**	**mAhr**	**Watt-Hours**	**Joules**	**Hours of ECG monitoring**
Miniature LiPo	3.3	138	0.45	1639	24
2 x Alkaline AA	3	1100	3.3	11880	179
2 x Alkaline AAA	3	2100	6.3	22680	342

Table 4.7: Device autonomy in ECG mode.

5

Concluding Remarks

5.1 Introduction

This PhD thesis has proposed a model-based approach to the energy-aware development of Cyber-Physical Systems (CPSs). This approach aims at considering all the different subsystems that compose a CPS when using an energy-aware design process and, therefore, it is described as holistic. This approach has been formulated through a number of methodological additions and enabled by the extension of modelling tools and the proposal of modelling guidelines to use them. Additionally it has been demonstrated in a practical setting through two case studies.

This chapter summarizes the different contributions presented throughout Chapters 3 and 4 and evaluates them against the different dimensions introduced in Chapter 1. Finally this chapter outlines future work areas. Some of these areas will be addressed in the follow-up research project INTO-CPS[1].

5.2 Research contributions

In this thesis eight research contributions have been presented. These are grouped under three different categories: Methodological Additions, Methods and Tools Improvements for Subsystems Modelling and Application to Case Studies. An overview of the contributions and the relation among them is provided in Figure 5.1. Each contribution has been given a short name for easier identification. The methodological additions presented in Contribution 1 (yellow block) are applied in the modelling of the different subsystems that compose the CPS. Each contribution that has led to improvements of modelling tools and techniques for the different subsystems (green blocks) are grouped in the dashed green box. This comprises contributions 2 to 6. Finally, these techniques have been used as explained in the methodology section in

[1] The official INTO-CPS project website is: `http://into-cps.au.dk/`

the two case studies (blue blocks), therefore constituting contributions 7 and 8. These are grouped in the dashed blue box.

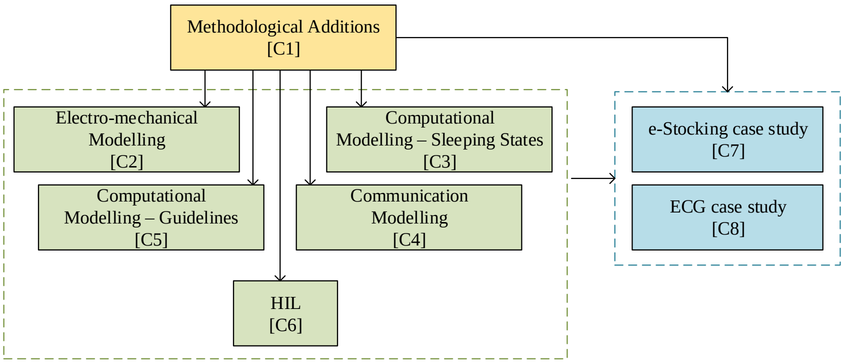


Figure 5.1: Overview of the contributions presented in this PhD thesis.

5.2.1 Methodological additions

Chapter 2 introduced a model-based lifecycle for CPS development that, as opposed to the traditional V lifecycle makes use of modelling techniques to support the different engineering activities. This model-based lifecycle has been taken as a starting point to formulate a number of additions over it, so it is possible to take energy consumption into consideration from a System-Level perspective. This constituted the first contribution, which was formulated as:

Contribution 1. Additions to the model-based engineering V lifecycle so it is possible to address the energy consumption through a holistic heterogeneous modelling approach.

These additions have been presented in section 3.1 and provide indications to the engineers or engineering teams using this methodology on how to apply the modelling techniques presented and how to use the results produced.

5.2.2 Methods and tools improvements for subsystems modelling

The second group of contributions contains the modelling tool improvements and modelling guidelines to represent and analyze the energy consumption. From the electromechanical perspective the modelling tool chosen to represent the physical aspects (20-Sim) was capable of representing energy consumption. This means that no tool modifications were necessary and therefore the contribution was focused on application guidelines:

> **Contribution 2.** Guidelines for the representation of energy and power consumption in electromechanical subsystems using co-simulation.

From the computation point of view the VDM-RT modelling method was extended to enable the representation of different operational modes. This was presented in section 3.3 together with modelling patterns to illustrate its application. This led to the following contributions:

> **Contribution 3.** Extension of the VDM-RT language for the representation of energy and power consumption in computational subsystems by enabling sleeping states.

> **Contribution 4.** Guidelines for the representation of CPU low power states at an abstract level, that corresponds to the way real CPUs operate.

From the communication point of view, the VDM-RT modelling method incorporated the necessary constructs to simulate small-scale networks. However, no clear guidelines to simulate energy consumption had been provided before. Section 3.4 addressed this aspect and presented Contribution 5:

> **Contribution 5.** Guidelines for the representation of energy consumption in communication using the distributed aspects of VDM-RT, as well as the representation of small-scale networks at an abstract level.

Finally, section 3.5 introduced Contribution 6, which presented an initial approach to HIL simulation of DE models developed in VDM-RT. This elaborated on the necessary interfaces and additional architectural extensions to the Overture tool laying the basic foundation over which more advanced HIL simulations could be built. Additionally, interaction with digital signal acquisition hardware was prototyped so automatic measurements were possible. This is summarized as Contribution 6:

Contribution 6. Extension of the Overture platform to partly support Hardware In the Loop simulation, therefore facilitating the combination of models with partial system realizations.

5.2.3 Application to case studies

The methodological additions together with the modelling guidelines and tool improvements have been applied to two case studies, that were presented in sections 4.2 and 4.3. This lead to the following contributions:

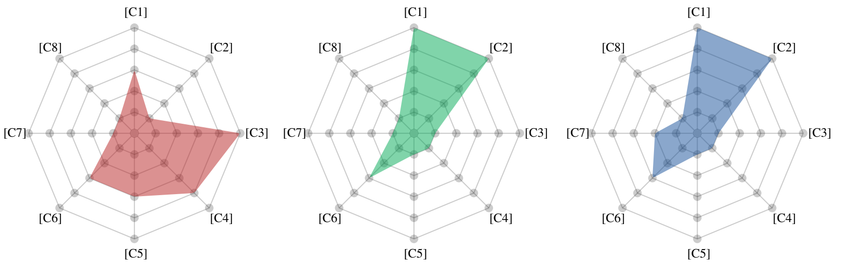
Contribution 7. The application of the methodological additions to the design and implementation of a compression stocking to treat leg-venous insufficiency.

Contribution 8. The application of the methodological additions to the design and prototyping of a wearable heart monitor.

Both case studies complemented each other regarding the kind of subsystems that were more demanding from the energy perspective. The first case study was more energy demanding on the electromechanical side while the second was more energy demanding on its computation and communication side. Additionally the first case study was based on a research project aiming at creating a real product, and presented more complex challenges if compared with the second one.

5.3 Evaluation of the research contributions

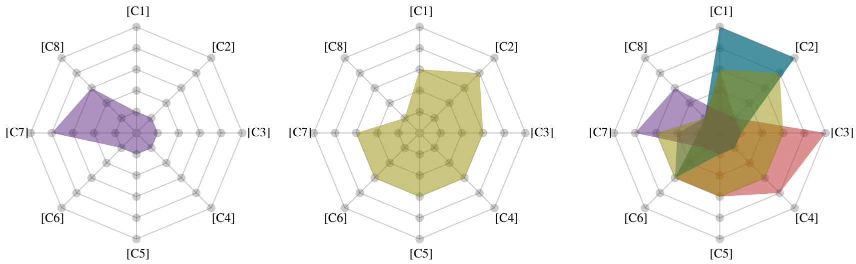
The following sections evaluate each of the research contributions presented in Chapters 3 and 4 and summarized above. This evaluation is performed against the different dimensions that were introduced in Chapter 1. An overview of the evaluation results is provided in Figure 5.2.



(a) Improvement of Modelling Methods and Tools

(b) Better System-Level Representation of CPS

(c) Collaboration Between Engineering Disciplines



(d) Application to Real Case Studies

(e) Shortening of Development Times and Reduction of Costs

(f) Combined Overview

Figure 5.2: Relation between contributions and evaluation criteria.

5.3.1 Improvement of modelling methods and tools

In order to model energy consumption in electromechanical components a number of modelling guidelines for the creation and instrumentation of CT models have been proposed. Even though the 20-sim environment was capable of analyzing energy consumption in this domain, clear modelling guidelines to do it have not been proposed previously in a co-simulation setting. These guidelines and their application in the creation and analysis of co-models allows to analyze effectively energy consumption in physical processes and the impact of different control strategies.

In order to support CPUs with sleeping operational states the VDM-RT scheduler and language have been modified. Additionally, guidelines to use the language extensions have been provided. Prior to this contributions the energy consumption in computation have not been addressed in a VDM-RT setting. The accuracy of these models is coarse grained and this could be a limitation if the intention is to conduct energy analysis at very low levels of abstraction. On the other hand it is adequate to conduct trade-off analysis between different sleeping policies.

Energy consumption in communication has been covered by Discrete Event models created in VDM-RT following the modelling guidelines and patterns proposed in Contribution 5. These take as starting point ready made commercial off-the shelf communication interfaces and protocols and the measurements of energy consumptions conducted over them. After that energy consumption analysis is based on those and the amount of data that is transmitted and received. This approach is valid to conduct analysis at a high level of abstraction but weak if the purpose is to design or compare different low-level communication protocols (for example Medium Access Control protocols). Additionally, due to the tools and their scalability problems this is limited to small-scale networks. A possible way to tackle these aspects is to delegate the simulation of more complex scenarios to other modelling and simulation tools such as NS2 or OMNeT++ [16].

Additionally an initial approach to HIL has been presented in Contribution 6. This is new within the VDM-RT community but not within embedded systems in general, since HIL has existed for decades. However this has served as a first step towards the usage of HIL in VDM-RT simulations over Overture and will be continued as described later in future work. Since the HIL-Overture techniques are at an early stage and their applicability is limited, they have been assessed with a low score if compared with the other contributions.

5.3.2 Better System-Level representation of CPS

To our knowledge no previous work exists that has formulated a holistic model-based approach to the analysis of energy consumption in CPS. Therefore, Contribution 1 scores high when evaluated in this dimension, since we consider this a concept that could potentially be extended and applied by other researchers working on methodologies for CPS development.

Contributions 2-6 have shown that existing modelling tools can be extended in order to give a more complete and holistic representation of CPS. In this sense they have enabled the better system-level representation evaluated in this dimension.

On the electromechanical side, Contribution 2 has not explicitly addressed energy harvesting as a physical process that could be simulated together with the physical or mechanical aspects of the system and this could be a key aspect in some energy-constrained systems.

Contributions 3 and 4, which addressed energy consumption in computation, can effectively represent energy consumption of different CPU operational states, however, they are limited when studying Dynamic Frequency Scaling energy saving techniques. At this moment the operating frequency of the VDM-RT CPUs is fixed before the simulation starts and cannot be changed at runtime. Additionally, the representation of different embedded Real-Time Operating Systems is limited to what the VDM-RT scheduler supports.

On the communication side, Contribution 5 has effectively addressed energy consumption in communication and provided guidelines to represent small-scale networks. This kind of simulation is limited by the VDM/Overture tool and more complex analysis is not possible without using other specialized network simulators.

Despite these limitations we believe that the additions to the language and the methodological guidelines proposed can effectively contribute to a better system-level representation, that is good enough for both reasoning about system architectural properties, functionality and energy consumption at a high level of abstraction.

5.3.3 Collaboration between engineering disciplines

The methodological additions presented in this PhD thesis assumes that a model-based approach is applied by either a single engineer with sufficient knowledge in electromechanical, computation and communication (possible but not common) or by a team of engineers that belong to different disciplines and therefore are experts in different fields. It is in this latter case where the approach proposed in this PhD thesis brings the additional value of fostering collaboration between the different disciplines and making them aware of the impact of their design decisions with regard to energy consumption in other system aspects.

In this sense, Contribution 1 and Contribution 2 score fairly high since they are explicitly focused on making this kind of interaction possible and because they are based on technologies that enable it.

Contributions 3 and 4 are very focused on computation alone and the interaction with other disciplines could be reduced to a basic specification of real-time deadlines that the system has to abide as well as an energy budget figure. Therefore both contributions score low in this dimension.

Although Contribution 5 is focused on communication it could potentially also be used with the modelling facilities provided by the VDM-RT to study trade-offs between computation and communication. Therefore we consider to some extent that collaboration between disciples is enabled.

Finally, the HIL proposed in Contribution 6 could definitely enable collaboration between disciplines and teams if physical prototypes developed by other teams could be integrated with simulations of subsystems developed by engineers from other disciplines. In that sense the HIL technology and its application in Overture/VDM-RT simulations has a promising potential. Due to the fact that this work is at an early stage and that the concept is not original this contribution is not assessed with a high score in this dimension.

5.3.4 Application to real case studies

Contributions 7 and 8 have been focused primarily on applying the methodological guidelines presented in contributions 1 to 6 in two different case studies. Both case studies are realistic and complex enough so the application of a methodology like the one proposed in here makes sense. Therefore, contributions 7 and 8 score high in this dimension but not with a top score. This is due to the fact that these case studies are not as complex as industrial cases of high scale that are tackled by teams of several engineers.

Contributions 1 through 6 score low in this dimension since they are focused on providing the methods and tools to conduct the application carried out later rather than constituting complex case studies by themselves. On the other hand these contributions have also been validated against smaller examples prior to their application to the case studies and therefore they do not score with a zero.

5.3.5 Shortening of development times and reduction of costs

The methodological additions framed in Contribution 1 score high in this dimension and they represent the model-driven engineering vision to CPS design presented in this PhD thesis. This assessment is supported by the application results in the case studies presented alone. The application of these additions to the e-stocking case study allowed providing relevant input to the mechanical engineers developing the compression mechanism that resulted in a less energy demanding solution. Additionally it allowed to evaluate different compression mechanisms without having to prototype them. In the heart monitor case study it has been possible in a simulated environment to determine the energy consumption of different computational and communication strategies without having to realize them either. Based on these experiences it can be concluded that these methodological additions have the potential of shortening development times since a) prototyping is replaced to some extent by modelling and b) engineers from different disciplines are able to integrate subsystem models earlier in the development process. This allows the early identification of System-Level problems as well as the early analysis proposed in the thesis. This has a direct impact on the reduction of development times and costs.

Contributions 2 to 6 score lower in this dimension since they do not explicitly contribute to development times shortening, however they are considered as enablers and therefore do not score with a 0.

Contributions 7 and 8 are case studies and as such they have not been considered in this dimension.

5.4 Hypothesis validation

Throughout this PhD thesis an number of contributions have been presented with the purpose of addressing different aspects of the hypothesis presented in section 1.4. These contributions, that have been evaluated above, have scored different in the evaluation dimensions. The combination of the contributions, shown in Figure 5.2 chart f, shows that overall progress has been made and that the evaluation results are sufficient to validate the hypothesis of this PhD.

5.5 Future work

5.5.1 Modelling languages and tools

From the electromechanical perspective it would be desirable to extend the current 20-sim toolboxes with a library that facilitates the representation of energy harvesting subsystems. Additionally, it could be relevant to incorporate battery models that take battery degradation into consideration. This would enable the simulation of the system performance in the long run.

On the computational side it would be relevant to incorporate more detailed performance models of concrete hardware platforms in the Overture environment, so more detailed VDM-RT models could be created without the need of performing partial prototyping and measurements.

On the communication side further exploration of computation versus communication trade-offs with other case studies could better illustrate the potential of the VDM-RT language in order to represent these kinds of logical problems, their impact on energy consumption analysis and how they could be addressed. Additionally, the platform could be combined with a network simulator in order to conduct more specialized Designed Space Exploration [22].

The work on HIL should be extended so a more robust and versatile solution is created and incorporated in the Overture tool. This would require to continue the extension of the software components developed for Overture as well as the integration with a concrete hardware platform. Ideally a well defined communication protocol between the external hardware and Overture should be made available so interfacing other kinds of embedded platforms would be simplified. The possibility of conducting time measurements over both hardware and software realizations should also be explored further as well as how the automatic validation of time conjectures [31, 98] performed over these measurements can help on validating functionality that could, ultimately, have an impact on energy consumption.

Finally it could be beneficial to consider the combination of high-level VDM-RT models with lower level SystemC [83, 9] models of hardware components, which are used extensively when designing embedded hardware platforms [37]. This could be interesting in case the system would incorporate embedded hardware that could be implemented in an FPGA[2]. In the same manner it could be beneficial to enable a combination of VDM-RT models with an environment capable of simulating analog electronics accurately, such as LTspice[3]. Note that here we are considering combination of models and not a full blown co-simulation, since it might not be necessary in order to carry out a high-level energy consumption analysis.

5.5.2 Code generation

Code generation could be an interesting aspect to exploit in order to facilitate the transition from models to subsystems realizations. Previous work exists in this direction for the VDM-RT modelling language [58] so it is possible to generate both C and C++ software implementations. This could potentially have benefits when it comes to realizing software solutions modelled in VDM-RT, that could be electromechanical controllers, computation models or communication protocols. The 20-sim platform through its extension

[3] FPGA stands for Field Programmable Gate Array. An FPGA [79] allows to implement custom hardware components through hardware modelling language such as VHDL [82, 94], VERILOG [81] and higher level languages such as SystemC [9].

[3] The official LTspice website is: `http://www.linear.com/designtools/software/`

20-sim 4C[4] [64] does incorporate C code generation so controllers can be synthesized and deployed in a variety of platforms, being the most relevant Raspberry Pi[5] (an embedded linux platform), Arduino[6] (typically an 8-bits microcontroller platform but other configurations are possible) and Bachmann PLCs[7] (industrial grade automation platform). Such code generation could be oriented towards control and regulation but it would be worthwhile exploring how it could be exploited and combined with the VDM-RT code generator when developing CPSs. Furthermore, the performance and degree of optimization of the generated code and when to code generate from one tool or the other should be an aspect to be covered in the future.

Finally, VHDL code generation in order to automatically realize hardware from computer models should be covered as part of the code generation work. The compilation of models to VHDL could enable the production of highly specialized hardware blocks with performance orders of magnitude higher if compared with their software counterparts. Additionally, the advent of low cost and accessible System on Chip (SoC) solutions would make this aspect more relevant than ever in the the coming future and has the potential of lower power consumption further.

5.5.3 Case studies

In order to determine the limits of the methodological additions presented in this thesis it would be worthwhile testing it in an industrial setting and preferably with more than one engineer using it. Additionally it would be beneficial if the engineers would belong to different disciplines. This would give a more practical insight into the application of model-driven design and demonstrate in practice how the methodology allows to take energy consumption into consideration when designing CPSs in a collaborative fashion.

[4] The official 20-sim4c website is: `http://www.20sim4c.com/`
[5] The official RaspberryPi website is: `http://www.raspberrypi.org/`
[6] The official Arduino website is: `http://www.arduino.cc/`
[7] The official Bachmann website is: `http://www.bachmann.info/en/`

Part II

Publications

6

A Holistic Approach to Energy-Aware Design of Cyber-Physical Systems

The paper presented in this chapter is under review.

[P54] José Antonio Esparza Isasa and Peter Gorm Larsen and Finn Overgaard Hansen *A Holistic Approach to Energy-Aware Design of Cyber-Physical Systems*. Submitted to the International Journal of Embedded Systems.

The content of this chapter has been excluded due to copyright restrictions, but can be obtained though the respective publisher.

7

Energy-Aware Design of Embedded Software through Modelling and Simulation

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P55] José Antonio Esparza Isasa, Peter Gorm Larsen and Finn Overgaard Hansen *Energy-Aware Design of Embedded Software through Modelling and Simulation*. Proceedings of the SYRCoSE 2014 Symposium. May 2014.

Energy-Aware Design of Embedded Software through Modelling and Simulation

José Antonio Esparza Isasa⋆, Peter Gorm Larsen⋆ and Finn Overgaard Hansen†
Department of Engineering⋆, Aarhus School of Engineering†
Aarhus University
Aarhus, Denmark
{jaei,pgl}@eng.au.dk⋆, foh@iha.dk†

***Abstract*—We present a model-driven engineering approach that enables to take energy consumption into account during the development of embedded software. In this approach we address all the constituents of a typical modern embedded solution (mechanics, communication and computation subsystems) through the application of different modelling technologies. This makes it possible to evaluate the implications of different software and system architectures in the system's energy consumption. Additionally it facilitates the exploration of the design space without having to prototype each candidate solution. We also provide details on the application of this approach to the development of a medical grade compression stocking and the benefits this approach has brought to the project currently developing this system.**

I. Introduction

Modern embedded solutions are typically a combination of computing units, communication interfaces and mechanical subsystems and they can operate both autonomously or as part of a network. This makes embedded solutions heterogeneous systems that are very hard to design [1]. In addition many embedded systems are battery powered and therefore they present the added complexity of begin energy efficient while still fulfilling their operational requirements [2].

A possible tool to cope with complexity is the application of abstract modelling. Modelling can be used to represent the system at the highest level of abstraction. These abstract models can be progressively transformed into a concrete system realization [3]. This approach is known as model-driven engineering.

This paper presents a model-driven engineering approach to the design of these complex embedded solutions. This approach makes use of several modelling paradigms in order to represent different aspects of the system and makes special emphasis on the energy performance of different candidate solutions. The modelling activities proposed under this approach are conducted early in the development process and allow design space exploration without requiring physical prototyping. This reduces development time and cost and enables the repeatability of the experimental simulations. Additionally it provides a tool to design quality solutions and to make well-founded design decisions.

The reminder of this article is structured as follows: Section II describes the design approach to energy consumption proposed in this paper. Section III elaborates on HIL and system realization following this approach. Section IV describes how this technique is being applied to a case study and its preliminary results. Sections V and VI present future and related work. Finally, Section VII concludes the paper.

II. An holistic model-driven engineering approach to Embedded Systems design

The approach proposed in here aims at studying the energy consumption in the different subsystems that compose typical embedded solutions. A general representation of such a solution is presented in the SysML block diagram shown in Fig. 1 and its components described below:

- **Embedded Hardware:** represents the electronic hardware that supports the execution of software and possibly other components implementing additional logic in hardware.
- **Embedded Software:** represents the software that controls the operation of the rest of the components in the embedded solution.
- **Mechanics:** represents the mechanical components that are controlled by the system. Interfacing with the mechanical subsystem and the environment from the embedded side is conducted through sensors and actuators.
- **Communications Interface:** represents the communication hardware that makes it possible to establish links with other networked systems.

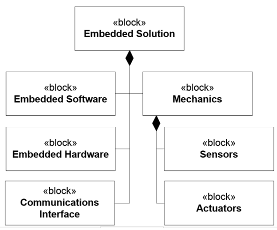


Fig. 1. SysML block representation of a typical embedded solution.

To design an embedded solution comprised by the components presented above so it satisfies its operational requirements is already a challenging task. To design it in such a way that it is low energy consuming is even more complex. In order to cope with the complexity associated to the design of this kind of systems we proposed the application of model driven engineering techniques that apply System-Level (SL) modelling. SL modelling aims at describing the system under design at the highest level of abstraction and to incorporate progressively system details in order to conduct system analysis. Such a model is gradually transformed into a final implementation. We propose the application of SL design taking into consideration all the energy consuming components in an Embedded Solution in a joint design effort. Additionally we propose Hardware-In-the-Loop as a way to combine executable models with partial system realizations in software and/or hardware running on target. Our approach is said to be holistic because it takes into consideration subsystems that, even though could seem to be unrelated at first sight, they all have an impact in total energy consumption. Therefore, our approach targets the mechanical subsystem, the communication interfaces and the computation logic executed in the embedded software and hardware. All these aspects are addressed in different specific ways in order to obtain energy consumption estimates, that can be used to study the total energy consumption. Additional details on each specific strategy are provided below.

A. Modelling mechanical subsystems

In order to address the design of mechanical subsystems we apply a co-modelling approach in which the engineers use a modelling paradigm able to represent Continuous Time (CT) phenomena and a second one in order to represent Discrete Event (DE) logic [4]. Controlled physical processes (plant) are best represented by using CT abstractions such as differential equations. On the other hand, the control logic that operates the plant is best represented by using logical formalisms.

We proposed a particular way of using this co-modelling approach so it is possible to take into account energy consumption during the design of mechatronic systems [5]. An overview of this approach is presented in Fig. 2. We take as starting point a CT-first methodology [6], in which the modelling starts by focusing on the mechanics of the system and carrying out the control logic modelling afterwards. The CT models are focused on the core functionality that has to be delivered and do not capture heat dissipation due to the conversion efficiency of the electromechanical devices. In case part of the system operation depends on its internal temperature, this could be represented as part of the CT models, since it is a physical process with impact on energy consumption.

Once the system operation has been described, the notion of energy is incorporated into the CT models in a phase that is called models instrumentation. The notion of energy is incorporated only in the CT side since the energy consumption in the DE side is typically negligible if compared with the first one. This instrumentation consist on basically monitoring the variables that have an impact on energy consumption and doing it in a way that does not affect the performance of the models. After this phase it is possible to co-execute the models and produce a number of energy consumption estimates, that can be used to perform trade-off analysis between the different modelled solutions. In case the energy consumption estimations are fed back to the DE model it is possible to use this approach to model energy-aware system operation, meaning that the system can feature different operational models that are switched among depending on the energy consumed. In principle this approach could be applied by using any tool that supports the co-simulation of DE and CT models.

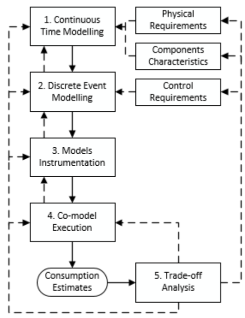


Fig. 2. Overview of the co-simulation based methodology.

In this work we have used the co-execution environment DESTECS/Crescendo [7], [8]. This environment combines the tools Overture [9] and 20-Sim[1]. Overture incorporates an interpreter for the DE modelling language VDM-RT [10]. This language is best suited to represent the control logic supervising the mechanical components and therefore it is used for DE modeling. 20-Sim incorporates a numerical engine that evaluates differential equations. Additionally it supports abstractions built on top of differential equations in the form of bond and block diagrams. This tool is best suited to represent the mechanical side of the system (CT side). The DESTECS environment synchronizes the co-execution of the models providing a common notion of time for both models. Additionally it allows the specification of a number of controlled and monitored variables between the VDM-RT and the 20-sim simulations so the supervision of the plant is possible. Additional details on how DESTECS and the process presented above have been applied in a concrete case study will be provided in Section IV.

B. Modelling computation subsystems

Modern microcontrollers can switch between different operational modes in order to reduce energy consumption. These operational modes temporarily disable the CPU and certain peripherals in order to achieve such a reduction. Switching

[1]20-Sim official website: `http://www.20sim.com`

between operational modes takes time that might have an impact in the real-time performance of the system under study. An example of different power modes can be seen in Fig. 3. In this case the processor features two low power operational states: Hibernate and Sleep. Sleep allows a fast CPU wake-up based on internally or externally generated events. Hibernate requires additional time to wake-up the CPU and it can react only on external events. However it can lower the energy consumption even further. The CPU controls from software how to switch between these modes.

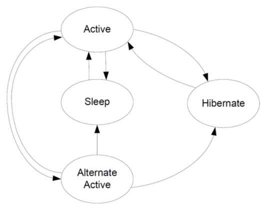


Fig. 3. CPU states in an implementation of a Cortex M3 ARM processor.

In order to explore the application of the different sleep modes, software strategies and architectures required for low power operation we propose the application of the modelling language VDM-RT. This language incorporates the abstraction **CPU** that represents an execution environment in which parts of the model can be deployed. Besides representing the computational support it also incorporates a real-time operating system layer. Logic running in VDM-RT CPUs can represent single or multi-threaded software implementations as well as dedicated hardware blocks depending on how they are configured [11]. However, VDM-RT CPUs do not feature the notion of low power consuming states since they are always active and therefore able to perform computations. An initial approach to overcome this limitation was a design pattern structure that regulated the access to the CPU as a resource by the logic running on it depending on the state of a flag [12].

As a way to overcome this situation we have extended the VDM-RT language by adding two new constructs to manage CPU states: **CPU.sleep()** and **CPU.active()** [13]. The addition of these two new constructs implied the modification of the VDM-RT interpreter and the scheduler built into the Overture platform. In addition to these extensions we proposed specific ways to use the **sleep** and **active** operation so the models can represent accurately the low power operation of real platforms. These templates show how to model a CPU wake-up based on an interrupt triggered by externally generated events and based on internal sleep timers. Model simulation produces a log file that registers in which states the CPU has been operating and for how long. This information together with the electrical characteristics of the CPU under consideration allows to represent the power consumption of the device over time. The integration of this curve over time results on the total energy consumption on the computational side.

The application of a modelling-based approach to the computational side of the system brings a number of advantages to the design of the software. Besides the obvious case of exploring different sleeping policies for a single CPU without having to protototype them, more complex cases in which several CPUs are involved can be explored. This is especially relevant if the CPUs have to communicate in order to satisfy system requirements.

C. Modelling communication subsystems

The VDM-RT modelling language incorporates the abstraction **BUS** that allows to communicate the **CPU** processing nodes introduced in the previous section. This abstraction can be used to represent point-to-point communication between CPUs in a static way. Communication performed over VDM-RT BUSes is assumed to be error-less, so any kind of communication problems such as information (packet) loss has to be modelled on top. At this point the BUS abstraction does not incorporate any notion of energy consumption during communication.

1) Modelling network topologies: We propose the application of a design pattern structure to overcome some of these limitations. Our initial approach takes as an example the communication in a wireless context but it could easily be extended to a different one. In Fig. 4 this structure is presented through a UML diagram. The general idea behind this pattern is to create a star topology network in which each networked embedded system is connected to a central component, that runs a simulated transmission medium. This structure is applied in VDM-RT by using CPUs to represent each networked device as well as a central component simulating the medium. Finally buses are communicating each device model with the medium model. In this way there is no direct connection between the individual CPUs representing the devices and any transmission goes through the simulated wireless medium. By using some of the VDM abstractions one can easily establish relations between the CPUs in the simulated wireless medium to represent whether a communication between two nodes is possible or not. Analyzing these "connection maps" during model execution is especially easy due to the expressiveness of the VDM language and it can be accomplished by using map comprehensions. If model simulation time is a concern the connection maps can be translated into a look-up file in which it is explicitly stated the relation between all the networked elements. This structure solves the initial problem of representing a realistic topology of a small scale embedded network in VDM-RT.

2) Introducing the notion of energy consumption: In order to represent the energy consumption we focus on modelling the operational state of the communications interface of the embedded device. We consider operational states the different modes in which the communication interface can be working, typically: transmitting, receiving or deactivated. For each mode the manufacturer provides an average power consumption

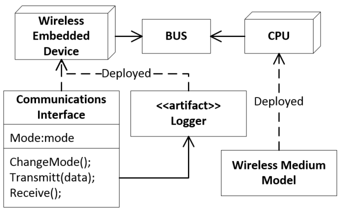


Fig. 4. Design pattern structure to represent wireless communication.

figure that can be used in the VDM-RT models. Changes among these operational modes are logged during model simulation and analyzed when it has been completed. Based on the transitions between the states and for how long the device has stayed on those states one can calculate the evolution of the power consumption over time and hence the total energy consumption during system operation.

Once the notion of energy consumption and the possibility of modelling different network topologies have been facilitated, it is possible to conduct the analysis of communication related problems. Some of these include but are not limited to, routing algorithms, network services, latencies or time synchronization between nodes. All these factors could be analyzed against energy consumption in order to get estimates that would allow an energy aware design of the communication subsystem, including communication software as well as, to some extent, hardware.

One of the advantages of using this structure is the clear separation between the connection map representing the network topology and conditions and the individual networked elements, even though the simulation of both is conducted in the same modelling environment. The main disadvantage of this approach at first sight are the limitations regarding the number of networked elements. We consider this approach valid only for small scale networks. However additional work is necessary to establish its practical limitations.

The approach to communications modelling proposed in this section is conducted only in VDM-RT without involving any other modelling paradigm. A co-simulation approach could be interesting to represent mobile communication nodes or a changes in the environment in which the network is deployed.

III. System realization and Hardware In the Loop

The approach presented above aims at tackling the design problems through modelling and simulation however, at some point, the system has to be realized. Given the fact that a strong emphasis has been placed on the modelling of the system it is desirable that the models created are used as much as possible during the system realization phase. This could include the combination of partial system realizations with models, allowing the co-execution of models with system realizations. The approach that we propose in this work is exemplified in Fig 5. In this diagram we show an initial VDM model of a system that executes a three phases algorithm in which data is acquired, processed and finally an output is provided.

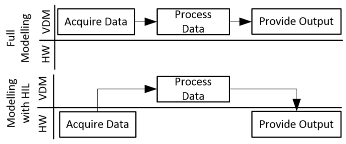


Fig. 5. Overview of the co-simulation based methodology.

We have applied this principle allowing the combination of VDM executable models running in a Workstation with actual components implemented in a Device Under Test (DUT) [14]. These components can be both hardware and software components. An overview of this Hardware In the Loop setup is presented in Fig. 6. In addition to the components mentioned above the system incorporates a Stimuli Provider able to simulate external inputs and a Logic Analyzer able to monitor the evolution of different logical signals. The VDM execution environment is able to interface the Logic Analyzer that can measure the time it takes to execute system realizations running on the DUT. This time figures can be manually incorporated into the VDM model and therefore increasing the fidelity of the model simulation results.

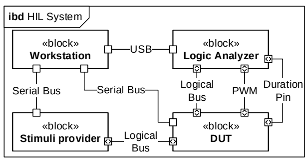


Fig. 6. SysML Internal Block Diagram showing the hardware connections to the DUT.

IV. Application and Preliminary Results

The approach proposed in this work is applied to the development of an intelligent compression stocking to treat leg venous insufficiency. This stocking is shown in Fig. 7 and it is composed of: an inner stocking (1), an inflatable stocking responsible for delivering the required compression levels (2), a pneumatical circuit composed of valves, pumps and a manometer (3), and an embedded system implementing the control logic and interfacing hardware and integrating a Bluetooth-based communication interface (4). This portable device is battery-operated and it is required to work for at least 14 hours. A complete description of this device can be found in [15]. As it is explained above this device is composed of mechanical, computational and communication subsystems that have to be energy efficient so the device autonomy requirements can be met.

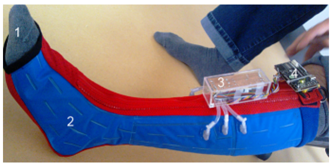


Fig. 7. The medical grade compression stocking.

A. Modelling of the Compression Principle

In order to study the mechanical subsystem we have applied the modelling process presented in Section II-A. The modelling of the mechanical system by itself was already beneficial for the project since it allowed us to gain a thorough understanding of the pneumatics and the physics behind the compression principle. Based on these models we were able to determine that a certain compression strategy was not feasible without having to prototype it. Additionally we were able to model different control software in VDM-RT and co-simulate its performance together with the mechanical models.

The analysis on both control strategies and mechanical pneumatic configurations, provided a number of energy consumption estimates that helped on deciding which system configuration was optimal. These suggestions had an impact on the system realization and introduced improvements on the software that increased its energy efficiency.

B. Modelling of the Embedded Software

We have applied the modelling techniques presented in Section II-B in order to explore two different embedded architectures in a concrete scenario: the regulation. The air pressure level in the air bladders have to be monitored periodically and kept at certain levels so proper compression is delivered to the limb. Through the regulation process the controller reads a manometer, compares the value retrieved against the expected one and depending on this triggers the pump or vents the bladders accordingly. This logic is implemented as a software component and requires the CPU to be active. However and depending on the kind of sensors that are used the CPU can be sleeping for a longer period of time. We have used VDM-RT modelling to study the energy consumption of two different kind of sensor configurations: the first one uses smart sensors that wake up the CPU in case an overpressure event occurs and the second one uses passive sensors that require a poll from the CPU in order to provide a reading. In the first case the CPU presented a lower power consumption than in the second case, since the sensors could run independently from the CPU. In the second case the CPU power consumption was higher because it required periodic wake-ups in order to check the sensors, which were not running independently. These results were expected since this was a simple case, however the purpose of applying the technique in this case was to show the modelling principle in a simple case study.

The predictions provided by these simulations where confirmed by measurements conducted on system realizations for both architectures with a fidelity of up to 95% [13].

C. Modelling of Communication

The modelling of the communication system remains as future work. A complete overview of the different communication scenarios in which this device can operate is presented in [16]. The intention is to model the critical scenarios in which the device is running on batteries. Based on these models we aim at making energy consumption estimations and evaluating computation vs. communication trade-offs.

V. Future work

We are planning to extend the work presented in here so the energy consumption analysis of the communication is also possible. Additionally we are planning to apply the analysis of energy consumption in computation in a more complex situation and possibly combining it with the communication, therefore being able to represent and analyze computation vs. communication trade-offs. We are also aiming at applying some of the modelling techniques presented in here to a second system so it is possible to make a stronger case for the SL energy-aware design approach for embedded solutions.

VI. Related work

The energy consumption problem in todays embedded solutions is well recognized and one of the top research priorities [2]. System Level design is also a well established technique especially in the hardware world, that it is seeing its expansion to other non-computing domains [3]. However and to our knowledge the application of System Level design with the particular intention of addressing the energy consumption problem during the development process through modelling and prototyping has not been formulated previously. Even though energy consumption in all subsystems has not been addressed in a single design effort previously significant work has been conducted in the individual fields of energy consumption in computation, communication and mechatronic systems.

Regarding energy consumption in computation, extensive work has been carried out in order to characterized different layers of abstraction. Some authors propose very accurate characterization of concrete computing platforms by taking into account energy consumption at the micro-architectural and the instruction level [17], [18]. This differs from the work presented in here in the fact that we consider an average power consumption figure within the active state of the CPU in order to obtain a coarse grained estimation over time. Other authors focus on the characterization of energy consumption at the service level [19]. In this case the authors consider the energy consumption at the OS level when the CPU is active. In our work we consider a single energy consumption figure for all the services provided by the OS, however, in case some of the OS operations result in a longer time having the CPU active, this will be considered under our approach as well even though the services have not been individually characterized. A more comprehensive review of techniques to study power consumption in computation can be found in [20].

As in the computation case, modelling of communication has been conducted at different levels of abstraction, ranging from energy consumption at the communications interface level to higher layers such as routing or application [21], [22]. Our work makes use of more simple power consumption models that, even though they are based on fixed average power consumption estimates are expected to provide sufficient detail to enable trade-off analysis of network algorithms.

Energy consumption in mechatronic systems is typically addressed as a particular application of well-established modelling platforms such as Matlab, Ptolomy or Modellica. Energy consumption has been typically considered just as any other design factor of industrial grade equipment and mechatronic components have not been traditionally considered together with embedded devices. However this situation is changing due to the increasing relevance of Cyber-Physical Systems [2].

VII. Conclusions

This paper has presented a modelling approach to energy-aware design of embedded systems. The preliminary application of this approach to a case study has enabled the exploration of different control algorithms, different hardware and software architectures and different mechanical configurations. This has made it possible to evaluate system performance against energy consumption early during the development process without needing a physical prototype. This work will be complemented in the near future with a study of energy consumption from the communication point of view. A more in-depth description of this work can be found in [23].

We hope that the approach proposed can inspire other researchers working with modelling applied to embedded system development and, to some extent, enact the application of modelling in the design of real embedded solutions.

References

[1] Thomas A. Henzinger and Joseph Sifakis, "The Embedded Systems Design Challenge," in *FM 2006: Formal Methods, 14th International Symposium on Formal Methods, Hamilton, Canada, August 21-27, 2006, Proceedings*, 2006, pp. 1–15.

[2] Banerjee, A. and Venkatasubramanian, K.K. and Mukherjee, T. and Gupta, S. K S, "Ensuring Safety, Security, and Sustainability of Mission-Critical Cyber-Physical Systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 283–299, 2012.

[3] S. K. Gupta, T. Mukherjee, G. Varsamopoulos, and A. Banerjee, "Research Directions in Energy-Sustainable CyberPhysical Systems," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 57 – 74, 2011.

[4] J. Fitzgerald, K. Pierce, and P. G. Larsen, *Industry and Research Perspectives on Embedded System Design*. IGI Global, 2014, ch. Collaborative Development of Dependable Cyber-Physical Systems by Co-modelling and Co-simulation.

[5] J. A. E. Isasa, F. O. Hansen, and P. G. Larsen, "Embedded Systems Energy Consumption Analysis Through Co-modelling and Simulation," in *Proceedings of the International Conference on Modeling and Simulation, ICMS 2013*. World Academy of Science, Engineering and Technology, June 2013.

[6] J. F. Broenink and Y. Ni, "Model-Driven Robot-Software Design using Integrated Models and Co-Simulation," in *Proceedings of SAMOS XII*, J. McAllister and S. Bhattacharyya, Eds., jul 2012, pp. 339 – 344.

[7] J. F. Broenink, P. G. Larsen, M. Verhoef, C. Kleijn, D. Jovanovic, K. Pierce, and W. F., "Design Support and Tooling for Dependable Embedded Control Software," in *Proceedings of Serene 2010 International Workshop on Software Engineering for Resilient Systems*. ACM, April 2010, pp. 77–82.

[8] J. Fitzgerald, P. G. Larsen, and M. Verhoef, Eds., *Collaborative Design for Embedded Systems – Co-modelling and Co-simulation*. Springer, 2014.

[9] P. G. Larsen, N. Battle, M. Ferreira, J. Fitzgerald, K. Lausdahl, and M. Verhoef, "The Overture Initiative – Integrating Tools for VDM," *SIGSOFT Softw. Eng. Notes*, vol. 35, no. 1, pp. 1–6, January 2010. [Online]. Available: http://doi.acm.org/10.1145/1668862.1668864

[10] K. Lausdahl, P. G. Larsen, and N. Battle, "A Deterministic Interpreter Simulating A Distributed real time system using VDM," in *Proceedings of the 13th international conference on Formal methods and software engineering*, ser. Lecture Notes in Computer Science, S. Qin and Z. Qiu, Eds., vol. 6991. Berlin, Heidelberg: Springer-Verlag, October 2011, pp. 179–194, ISBN 978-3-642-24558-9. [Online]. Available: http://dl.acm.org/citation.cfm?id=2075089.2075107

[11] J. A. E. Isasa, P. G. Larsen, and K. Bjerge, "Supporting the Partitioning Process in Hardware/Software Co-design with VDM-RT," in *Proceedings of the 10th Overture Workshop 2012*, ser. School of Computing Science, Newcastle University, 2012.

[12] J. A. E. Isasa and P. G. Larsen, "Modelling Different CPU Power States in VDM-RT," in *Proceedings of the 11th Overture Workshop 2013*, ser. Aarhus University, June 2013.

[13] J. A. E. Isasa, P. W. Jørgensen, and C. Ballegaard, "Modelling Energy Consumption in Embedded Systems with VDM-RT," in *Proceedings of the 4th International ABZ conference.*, July 2014.

[14] J. A. E. Isasa, P. W. Jørgensen, and P. G. Larsen, "Hardware In the Loop for VDM-Real Time Modelling of Embedded Systems," in *MODELSWARD 2014, Second International Conference on Model-Driven Engineering and Software Development*, January 2014.

[15] T. F. Jensen, F. O. Hansen, and J. A. E. et al., "ICT-Enabled Medical Compression Stocking for Treatment of Leg-Venous Insufficiency." in *International Conference on Biomedical Electronics and Devices (BIODEVICES 2014)*, March 2014.

[16] F. O. Hansen, T. F. Jensen, and J. A. Esparza, "Distributed ICT Architecture for Developing, Configuring and Monitoring Mobile Embedded Healthcare Systems." in *International Conference on Health Informatics (HEALTHINF 2014)*, March 2014.

[17] Mostafa E.A. Ibrahim and Markus Rupp and Hossam A. H. Fahmy, "A Precise High-Level Power Consumption Model for Embedded Systems Software," *EURASIP Journal on Embedded Systems*, vol. Volume 2011, no. 1, January 2011.

[18] Sheayun Lee and Andreas Ermedahl and Sang Lyul Min, "An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors," 2001.

[19] B. Ouni, C. Belleudy, and E. Senn, "Accurate energy characterization of os services in embedded systems," *EURASIP Journal on Embedded Systems*, vol. 2012, no. 1, p. 6, 2012.

[20] Unsal, O.S. and Koren, I., "System-Level Power-aware Design Techniques in Real-Time Systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1055–1069, 2003.

[21] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 272–287.

[22] R. Min and A. Chandrakasan, "A framework for energy-scalable communication in high-density wireless networks," in *Proceedings of the 2002 International Symposium on Low Power Electronics and Design*, ser. ISLPED '02. New York, NY, USA: ACM, 2002, pp. 36–41.

[23] J. A. E. Isasa, "System-Level Energy Aware Design of Cyber-Physical Systems," Department of Engineering, Aarhus University, Finlandsgade 22, Aarnus N, 8200, Tech. Rep., October 2013, available on-line at http://eng.au.dk/fileadmin/DJF/ENG/PDF-filer/Tekniske_rapporter/ECE-TC-16-samlet.pdf.

8

Embedded Systems Energy Consumption Analysis Through Co-modelling and Simulation

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P49] José Antonio Esparza Isasa, Peter Gorm Larsen and Finn Overgaard Hansen *Embedded Systems Energy Consumption Analysis Through Co-modelling and Simulation*. Proceedings of the International Conference on Modeling and Simulation (ICMS 2013) June 2013.

Embedded Systems Energy Consumption Analysis Through Co-modelling and Simulation

José Antonio Esparza Isasa, Finn Overgaard Hansen and Peter Gorm Larsen

Abstract—This paper presents a new methodology to study power and energy consumption in mechatronic systems early in the development process. This new approach makes use of two modeling languages to represent and simulate embedded control software and electromechanical subsystems in the discrete event and continuous time domain respectively within a single co-model. This co-model enables an accurate representation of power and energy consumption and facilitates the analysis and development of both software and electro-mechanical subsystems in parallel. This makes the engineers aware of energy-wise implications of different design alternatives and enables early trade-off analysis from the beginning of the analysis and design activities.

Keywords—Energy consumption, embedded systems, model-driven engineering, power awareness.

I. Introduction

POWER and energy consumption has become key factors to optimize modern embedded systems. It is very common to find embedded software controlling electronical and mechanical hardware forming a mechatronic system. Optimizing power and energy consumption in this kind of systems is especially complex, since it requires the cooperation of several engineering disciplines in charge of the different kinds of subsystems. In this paper we propose a cooperative modelling based methodology to take power and energy consumption into account early in the development process. This methodology takes into consideration both Discrete Event control software and Continuous Time electro-mechanical hardware from the modelling perspective and allows the evaluation of different control strategies to minimize power and energy consumption.

This paper continues as follows: In section II we present the modelling technologies that are used in our methodology, presented in section III. In section IV we show the application of this methodology in a concrete case study: the e-stockings project. In section V we discuss the limitations of this methodology and its advantages and disadvantages. In section VI we detail some of our planned work to improve and extend this methodology. In section VII we describe related work and finally section VIII concludes this paper.

II. Modelling technologies used

The methodology presented in this paper is based on two modelling languages (VDM-RT and 20-sim) and a tool to use them cooperatively (DESTECS):

VDM-RT: is an extension to the software modelling language VDM++ [1] and enables the modelling of real-time embedded control software. This modelling language is ideal to represent Discrete Event (DE) systems. VDM-RT models can be created and executed in the Overture tool [2].

20-sim[1]**:** is a physical modelling tool capable of representing electrical and mechanical systems among others. Modelling can be done by using bond graphs, iconic diagrams or differential equations. This modelling language and tool is best suited to represent Continuous Time (CT) systems. Standalone 20-sim models can be created and executed in the 20-sim tool [3], [4].

DESTECS[2]**:** is a co-simulation tool that integrates VDM-RT and 20-sim [5]. It communicates the VDM-RT interpreter with 20-sim and provides a common notion of time to synchronize the parallel execution of DE and CT models, now considered a single co-model. DESTECS also provides methodological guidelines to design mechatronic systems [6].

For illustration purposes in this paper we use the modelling language SysML [7].

III. Methodology

We propose the application of a methodology composed of five phases. An overview of this modelling process is presented in Fig. 1. These phases progress in a sequential manner, illustrated by a solid arrow between the phases. Additionally it is possible to iterate over the different phases depending on the modelling progress (illustrated by the dashed lines to previous phases).

In this methodology we take as input the Physical Requirements and Component Characteristics in the CT modelling phase and Control Requirements in the DE modelling phase (shown by dot-dashed arrows in the diagram). Additionally, we highlight the result of the Co-model Execution phase; Consumption Estimates, that serve as input to the Trade-off Analysis phase. In case the results of the Trade-off Analysis phase shows that it will not be possible to meet the energy and/or power consumption requirements, it is possible to revisit the previous phases and consider alternative physical realizations or different components.

J.A. Esparza, F.O. Hansen and P.G. Larsen are with the Department of Engineering, Aarhus University, Aarhus, Finlandsgade 22, 8200, Denmark. e-mail: jaei@iha.dk, foh@iha.dk, pgl@iha.dk.
Manuscript received January 31, 2013

[2] 20-Sim official website: http://www.20sim.com/
[2] DESTECS official website: http://www.destecs.org/

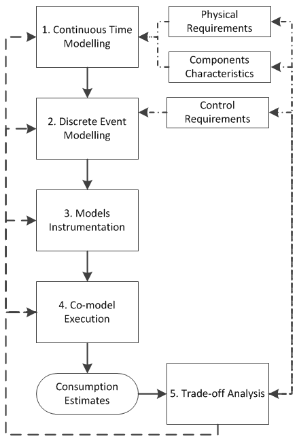


Fig. 1. Methodology overview.

A. Phase Description

1) Continuous Time Modelling: In this phase we model the system from a mechanical perspective. We take into account the physical requirements the system has to met as well as the components characteristics. At this point the physical model can be exercised with control signals to check that the physical model is appropriate.

2) Discrete Event Modelling: In DE Modelling we take the control requirements as input and focus on capturing the control logic. We will start with a simple control strategy, that exercises the mechanical model created in the CT Modelling phase. The purpose of this is to validate the physical models further before evaluating more complex control strategies. In case we find errors in the physical model, these can be corrected and iterate over the DE Modelling phase again. Once this additional validation is completed, we can model further more complex control strategies and iterate again over the CT Model to evaluate its behaviour.

3) Models Instrumentation: In this phase we incorporate the necessary interfaces and monitored variables to evaluate the energy consumption. The instrumentation will take place mostly at the CT model side but it leaves room for further modifications to the DE models as well. The instrumentation is performed by adding target variables as monitored in the physical simulator. Additionally, certain scaling and additional equations may be necessary. Since this depends highly on the mechanical/physical system under study, we cannot provide further details on this.

In order to measure the power consumption we propose the introduction of a power meter block per component in the CT model. Each power meter block will be responsible for computing the power consumption of that single component. Each power meter will take the component operating voltage and current as input. The calculation of the consumed power and energy will depend on the control signal provided by the DE model. Optionally, the DE model can take power and energy measurements into account at runtime with the purpose of modelling power and energy-aware operation at the component level. The calculated power consumption is represented as an output flow from the power meter for later processing in the CT model. The deployment of this block is shown by a SysML block diagram in Fig. 2. This diagram uses ports with arrows inside to represent continuous data flow (stream ports) and empty ports to represent discrete data communication (standard ports).

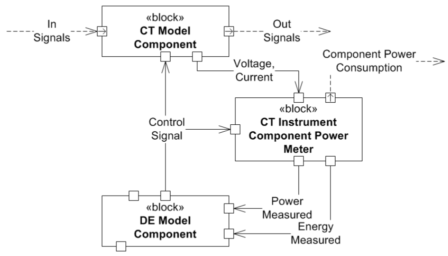


Fig. 2. Power Meter measuring component power and energy consumption.

Finally, and in order to compute the system total power consumption, we communicate all the component power meters with the system power meter. This block adds the power consumption logged by each component power meter and performs the integration of the power consumed over time in order to calculate the total system energy consumption. The deployment of this block is shown in Fig. 3.

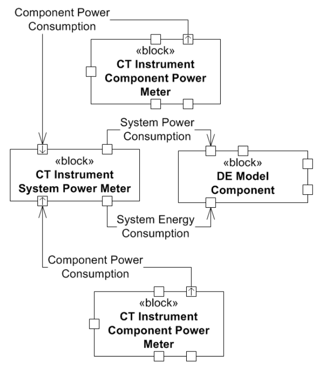


Fig. 3. System Power Meter measuring total power and energy consumption.

4) Co-model Execution: In this phase we execute the co-model and further study the results of the control algorithms on the mechanical system. Thanks to the model instrumentation it is possible to monitor power consumption over time and compare the impact of different control strategies to the energy consumption. In case we detect that additional modelling rework or instrumentation adjustment is necessary we can revisit any of the previous phases. As a result of this phase we will produce a number of system consumption estimates.

5) Trade-off Analysis: In this phase we evaluate the final energy and power consumption results and compare them against the system requirements. An example of the trade-offs we might encounter could be: energy consumption vs. accuracy, autonomy or size. Additionally we will be able to analyze the power and energy consumption of the individual components. In case the results we obtain through the trade-off analysis show that the requirements cannot be met with the solution considered, we can revisit the chosen components and reconsider their suitability. Additionally, the previous modelling phases can be revisited in an iterative fashion if we want to modify the models further.

B. Abstractions and model limitations

The purpose of the models is to get an overview of how the energy is spent on the physical side of the system and how different control strategies affect the energy consumption. The level of detail present in the models should be enough to perform trade-off analysis at the system level and therefore to achieve a *coarse-grained estimation*.

As part of the process of setting the level of abstraction, some of the physical components will be left out in the physical models. We propose to follow these steps in order to decided which components to incorporate in the models:

1) Identify the most energy consuming sensors and actuators and incorporate them in the CT model.
2) Model the control logic of the sensors and actuators detected in the previous step in the DE model.
3) Study sensors and actuators that have lower energy consumption than the ones detected in step 1. In case the frequency of use makes their energy consumption comparable to those identified in step 1, incorporate them into the models.
4) Model the control logic of the sensors and actuators identified in step 3.

Regarding the level of abstraction in the DE models, we propose to limit the DE modelling only to the control logic. In case the systems engineer is interested on performing additional modelling on other software functionality, we advice to create a separate VDM-RT model targeting only software related issues and creating a VDM-RT mock-up model representing the interaction with the physical world in a signal based fashion.

IV. The E-Stockings Case study

We have applied this methodology in the design of an electro-pneumatic compression stocking to treat leg venous insufficiency that is under development by the Ambient Assisted Living E-Stockings project[3]. This stocking is required to deliver a continuous compression gradient that ranges from 40 mmHg at the ankle to 20 mmHg below the knee. The stocking must reach the pressurized state and keep compression using as little energy as possible as it is operated by batteries.

[3]E-stockings official website: http://www.e-stockings.eu/

A. System overview

The stocking is composed of two independent air bladders around the leg that are inflated to build pressure and a special layer of textile that transfers the pressure to the leg with the proper gradient.

The preliminary mechanical architecture is composed off 3 electrically controlled pneumatic actuators:

Pump: able to maintain a constant airflow that is delivered to the air chambers around the leg.

Air distribution valve: able to direct the air flow to the first or the second air chamber. It also allows to vent the bladders individually.

Pass valve: able to lock the air bladder or open it when activated. Each air bladder has one pass valve.

These components have been arranged in such a way that makes it possible to keep the air in the bladders without requiring the constant consumption of energy. A SysML block diagram of the pneumatic subsystem is shown in Fig. 4.

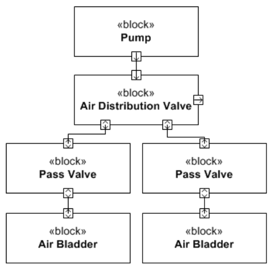


Fig. 4. SysML block diagram of the pneumatic subsystem.

Additionally, the stocking has two in-system pneumatic air pressure sensors to determine the air pressure in each of the two air bladders.

B. Application of the methodology

We have applied the methodology presented in section III to evaluate the consumption in the most power demanding scenario, the inflation of the air bladders.

1) Continuous Time modelling: We started by modelling the electro-pneumatic side of the system by using 20-sim blocks for the pneumatics domain. Additionally, we incorporated a transfer function modelled with a differential equation to represent how the pressure built in the air bladder is exerted through the textile to the patient leg. At this point we used fixed signals in the CT domain to test the model.

2) Discrete Event modelling: In this phase we started by modelling a dummy control algorithm with a simple *if-then-else* logical structure that made the pump active if the pressure was below the set-point or inactive if this was reached. Additionally it was possible to vent air if overpressure occurred. This simple algorithm made it possible to further test the CT models. Finally, we incorporated a more complex control algorithm: a PID regulator with two different set-points (one per air-bladder). The control algorithm is modelled as a periodic thread with a frequency of 5 Hz.

3) Models instrumentation: In this phase we instrumented the models so it was possible to keep track on the consumed power and energy by the electro-pneumatic system. We focused on the consumption of the pump and the valves since it was considerably above the rest of the components, so the instruments were incorporated only to monitor these. We used the pattern presented in section III-A3, making use of one power meter per component under study. Finally we communicated these component power meters to the system power meter.

4) Co-model execution: We ran the models under a common scenario, the inflation of the air bladders to reach the target pressure. We experimented with the dummy control algorithm and with the PID control algorithm under different configurations. We exported the logged data and created graphs for further analysis.

5) Trade-off analysis: Finally we analyzed the power and energy consumption results logged by the power meters and obtained different estimates for each control configuration. We were also able to conclude that a pure proportional controller with no integral or derivative part was sufficient to control the pneumatic system and make it stable with no overshooting. After analyzing the power consumption we realized that this could be reduced by rearranging the valves in the CT model. We iterated over the phases reworking the CT and the DE models and proposed an alternative architecture that features a considerable energy reduction. We present a more thorough study of the modelling and trade-off analysis results below.

C. Results and input for further development

1) Power consumption: After running the different control strategies modelled on the DE side with the same CT model, we have obtained four different power consumption estimates (one per DE strategy considered). System power consumption over time is shown in Fig. 5. Graphs a, b and c show the power consumption of the proportional controller with gains 2, 1 and 0.5 respectively. Finally, graph d shows the evolution of the power consumption under the dummy control algorithm. The dashed lines in each graph show the average power consumption in each case. The maximum average power consumption is 1.3 Watts, present when the dummy control is applied. The least power consuming control strategy is the proportional controller with gain 1, with an average consumption of 1.1 Watts. In all four cases the peak power consumption is 1.5 Watts.

2) Energy consumption: We have calculated the system energy consumption by integrating the power consumption curves presented in Fig. 5. Additionally we have paid special attention to the energy consumed by the pump under the different algorithms. These estimates together with the total time the system has been active in each case are shown in Table I.

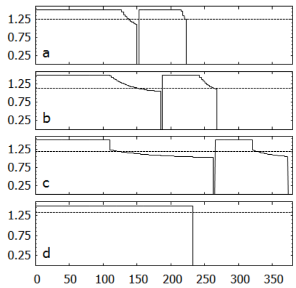


Fig. 5. System power consumption over time under different control algorithms for stocking compression. y axis: power in watts, x axis: time in seconds.

TABLE I
ENERGY CONSUMED UNDER DIFFERENT CONTROL ALGORITHMS

Control Algorithm	System Energy Consumption [Joules]	Time Active [Seconds]	Pump Energy Consumption [Joules]
p gain 2	323	222	103
p gain 1	367	267	102
p gain 0.5	472	372	102
dummy control	348	232	116

The pump energy consumption under the different proportional control strategies is practically the same, close to 103 Joules. The dummy control makes the most inefficient use of the pump, since it consumes 116 Joules to reach the same pressure level. At the system level it is evident that, even though the pump consumption was almost the same in the first three cases, there is a considerable difference in consumed energy between the proportional controllers (up to 149 Joules). This difference is due to the fact that the controller with a higher proportional gain remains active for less time, and therefore requires the valves that enable inflation to be active for a shorter period of time. The most energy demanding configuration is thus the proportional controller with a gain of 0.5, that requires up to 372 seconds to reach final pressure.

3) Control algorithms: Through the modelling activities we have deduced that it is not possible to drive the pneumatic subsystem in a completely proportional manner, something that is a requirement for a full PID control. It is possible to drive the inflation process of the air bladders in a proportional manner, but not the venting process. In case the system needs to vent air a considerable amount of energy is wasted. Using a dummy software controller is energy inefficient as well since it makes excessive use of the pump. This suggests that we should make use of a proportional controller for the inflation process and adjust the controller gain depending on the target inflation time.

4) Suggestions for mechanical modifications: With the current system mechanical architecture we need to keep running the pump, the air distribution valve and one pass valve, corresponding to the air bladder under inflation at the same time. This makes the power consumption peak at 1.5 Watts. We found out that it is possible to avoid the air distribution valve and cover its distribution functionality with a fixed air splitter and its selective air-bladder venting functionality with an additional pass valve per air bladder configured as sealed by default and connecting the air bladder to the exterior. This would make it possible to reduce peak power consumption to 1 Watt, a reduction of 33%. The total energy savings with this new architecture developed after model analysis are shown in Table II.

TABLE II
POTENTIAL ENERGY SAVINGS

Control Algorithm	System Energy Savings [Joules]	Reduction	Final System Energy Consumption [Joules]
p gain 2	111	34%	211
p gain 1	133	36%	234
p gain 0.5	186	39%	286
dummy control	116	33%	232

V. DISCUSSION

A. Model alignment and fidelity

The reliability of the results produced by this methodology depends on the accuracy of the models. In order to make reliable decisions we must align the models with the real components and we must use an accurate representation of the interaction with the physical environment. Modelling the component power consumption is an straightforward task that can be done by reading the component manufacturer specifications. On the other hand, modelling the interaction with the environment can be complex. In case this interaction is hard to represent, we advice to isolate it in a transfer function block so it is easier to refactor. We advice to build complex CT models incrementally: start by creating a simple yet meaningful representation of the physical interaction with the system and improve it in the following iterations. Even though the energy and power consumption estimates may not be reliable during the initial iterations, they would give the modeller a good grasp on how the components use energy and what potential improvements could be done.

B. Methodology applicability

This methodology is applicable to any system that makes use of electrical and/or electromechanical components. Additionally, this methodology can be applied to systems which are required to keep a constant or below a certain threshold power consumption, even though they have an unconstrained energy supply. Furthermore, this methodology can be applied to design systems that adjust their behaviour depending on energy and/or power consumption (energy or power-aware systems). This methodology does not consider at this point the evaluation of energy consumption in the software or in the communication side. However, these concepts can be easily evaluated by representing the computation and communication hardware units as blocks in the CT side.

C. Modelling vs. prototyping

Applying the model-based methodology proposed in this article brings important advantages to the development process when compared with a prototyping-based approach. This methodology requires an additional effort during the early stages of the project but allows the study of different system electro-mechanical architectures and control software in a more flexible manner. Development through prototyping would require constant measurements and complex data analysis after test runs. Applying modelling simplifies this task and eliminates other problems like measurement or data analysis errors.

VI. FURTHER WORK

We are planning to expand this methodology further so it takes energy consumption at the computation and communication levels into account. We are planning to test it further in additional case studies that represent typical applications in the embedded systems domain such as navigation systems and wireless sensor networks. We expect to experiment further with the possibility of power and energy aware system operation in these cases.

Regarding the e-stockings case study we are going to conduct further validation of the power and energy consumption estimates with a real implementation of both embedded control software and electro-pneumatic system. We are relying on this model to study other system aspects such as software-driven testing of the electro-mechanical components, filtering of sensor readings and software strategies to tolerate possible mechanical failures.

VII. RELATED WORK

Modelling applied to energy and power consumption is not a new idea and we can find numerous examples of related work in this area. However we are not aware of previous work in which modelling or co-simulation is applied at the system level, considering the system as a combination of control software, electronics and mechanics.

Modelling is typically applied at different levels of abstraction, some of them much lower levels than we target. For example Lee et al. [8] and Ibrahim et al. [9] propose the application of modelling to study the energy consumption at the processor instruction-level through processor specific models. Other authors like Park et al. propose the application of modelling at several levels of abstraction [10], but are still limiting the energy and power consumption analysis to the processor hardware. Celebican et al. focus on energy consumption of embedded system peripherals in [11], a valid approach to design System on Chip devices and going further than the work described above, but still limited to the analysis of electronic hardware.

The impact of software in energy and power consumption has also been taken into consideration in previous work. For example, Ouni et al propose an energy characterization of OS services in [12]. In this approach the hardware is not represented as accurate as in the work mentioned above. An intermediate methodology is followed by Vijaykrishnan et al., that propose a joint hardware-software approach to optimize energy consumption in [13]. However the authors use a simulated processor instead of a real one.

Wang et al. take the network perspective into account and propose models to study the energy consumption in embedded wireless nodes [14]. This approach enables a wider study of the interaction of the system with other systems but still does not tackle the control aspects.

Zhang et al [15] and Rao et al. [16] propose specific battery performance models and they highlight their relevance in mechatronic systems, but they do not elaborate on a concrete approach to use their techniques.

Finally, a related area of research is the application of modelling energy and power-awareness of high performance computing systems. Tiwari et al. use modelling targeting clusters with very high energy consumption in [17]. Even though modelling is applied in this scenario, the area of applicability is out of the scope of the work presented in this paper.

VIII. Conclusion

We have presented a new methodology to address energy and power consumption in mechatronic systems. This methodology is focused on the energy and power consumption of electro-mechanical components. We have applied this methodology to a case study and we are planning to conduct further validation of the energy and power consumption estimations. The results of applying this methodology have been positive and they have provided valuable input to the development phase of the final product, by pointing to concrete improvements in both software and mechanical sides that yield considerable energy savings.

References

[1] John Fitzgerald and Peter Gorm Larsen and Marcel Verhoef, "Vienna Development Method," *Wiley Encyclopedia of Computer Science and Engineering*, 2008, edited by Benjamin Wah, John Wiley & Sons, Inc.

[2] Peter Gorm Larsen and Nick Battle and Miguel Ferreira and John Fitzgerald and Kenneth Lausdahl and Marcel Verhoef, "The Overture Initiative – Integrating Tools for VDM," *ACM Software Engineering Notes*, vol. 35, no. 1, p. , January 2010. [Online]. Available: {http://dl.acm.org/citation.cfm?id=1668864}

[3] Controllab products, "http://www.20sim.com/," January 2013, 20-Sim official website.

[4] C. Kleijn, *20-sim 4.1 Reference Manual*, First ed. Enschede: Controllab Products B.V., 2009, ISBN 978-90-79499-05-2.

[5] J. F. Broenink and P. G. Larsen and M. Verhoef, and C. Kleijn and D. Jovanovic and K. Pierce and F. Wouters, "Design Support and Tooling for Dependable Embedded Control Software," in *Proceedings of Serene 2010 International Workshop on Software Engineering for Resilient Systems*. ACM, April 2010, pp. 77–82.

[6] Jan Broenink et al, "DESTECS Methodologogical Guidelines, Deliberable 2.3," DESTECS consortium, Tech. Rep., August 2012.

[7] Sandford Friedenthal, Alan Moore, Rick Steiner, *A Practical Guide to SysML*, First ed. Sanford Friendenthal: Morgan Kaufman OMG Press, 2008, ISBN 978-0-12-374379-4.

[8] Sheayun Lee and Andreas Ermedahl and Sang Lyul Min, "An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors," 2001.

[9] Mostafa E.A. Ibrahim and Markus Rupp and Hossam A. H. Fahmy, "A Precise High-Level Power Consumption Model for Embedded Systems Software," *EURASIP Journal on Embedded Systems*, vol. Volume 2011, no. 1, January 2011.

[10] Y.-H. Park, S. Pasricha, F. J. Kurdahi, and N. Dutt, "A multi-granularity power modeling methodology for embedded processors," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 4, pp. 668–681, 2011.

[11] Ozgur Celebican and Tajana Simunic Rosing and Vincent J. III Mooney, "Energy Estimation of Peripheral Devices in Embedded Systems," in *Proceedings of the 14th ACM Great Lakes symposium on VLSI*, ser. GLSVLSI '04, 2004.

[12] B. Ouni, C. Belleudy, and E. Senn, "Accurate energy characterization of os services in embedded systems," *EURASIP Journal on Embedded Systems*, vol. 2012, no. 1, p. 6, 2012.

[13] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye, "Energy-driven integrated hardware-software optimizations using simplepower," *SIGARCH Comput. Archit. News*, vol. 28, no. 2, pp. 95–106, May 2000.

[14] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, sept. 2006, pp. 286 –295.

[15] F. Zhang, Z. Shi, and W. Wolf, "A dynamic battery model for co-design in cyber-physical systems," in *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 51–56. [Online]. Available: http://dx.doi.org/10.1109/ICDCSW.2009.62

[16] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy-aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1109/MC.2003.1250886

[17] A. Tiwari, M. A. Laurenzano, L. Carrington, and A. Snavely, "Modeling Power and Energy Usage of HPC Kernels," in *Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, ser. IPDPSW '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 990–998.

José Antonio Esparza Isasa received his MSc. degree in Technical IT Engineering in 2012 from Aarhus University where he is currently working on his PhD. His main areas of interests are modelling and simulation of embedded systems and power and energy-aware systems.

Finn Overgaard Hansen is currently a professor at Aarhus University School of Engineering, where he is leading the embedded systems team. He received his M.Sc. degree at the Technical University of Denmark in Electronic Engineering in 1977. Has been working in industry from 1978-1984 as a software developer of embedded systems and has for 15 years been working at the Danish Technological Institute with development and dissemination of software methods to the Danish Industry. Returned to academia in 2001 with the task of establishment of a master degree in Information technology and building up a research program for embedded and distributed systems. His prime research interest is in architecture and design of software for networked embedded systems.

Peter Gorm Larsen is currently a professor at Aarhus University School of Engineering where he is leading the software engineering team.

After receiving his M.Sc. degree at the Technical University of Denmark in Electronic Engineering and Computer Engineering in 1988, he went to industry to bridge the gap between academia and industry. He later returned and did an industrial Ph.D. degree which was completed in 1995. He gave industrial courses all over the world, and had an industrial career until he decided to return to academia in 2005.

His prime research interest is to improve the development of complex mission-critical applications with well-founded technologies and in particular tool building that enables such development. He is the author of more than 90 papers published in journals, books and conference proceedings and a couple of books.

9

Modelling Energy Consumption in Embedded Systems with VDM-RT

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P50] José Antonio Esparza Isasa, Peter W.V. Jørgensen and Claus Ballegaard Nielsen *Modelling Energy Consumption in Embedded Systems with VDM-RT*. Proceedings of State Machines, Alloy, B, VDM, and Z 2014 (ABZ 2014), April 2014.

The content of this chapter has been excluded due to copyright restrictions, but can be obtained though the respective publisher.

10

Energy-Aware Model-Driven Development of a Wearable Device

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P56] José Antonio Esparza Isasa, Peter Gorm Larsen and Finn Overgaard Hansen *Energy-Aware Model-Driven Development of a Wearable Health Care Device*. 4th Symposium FHIES/SEHC June 2014. In Press.

11

Modelling Different CPU Power States in VDM-RT

The paper presented in this chapter has been accepted as a peer-reviewed workshop paper.

[P52] José Antonio Esparza Isasa and Peter Gorm Larsen *Modelling Different CPU Power States in VDM-RT*. Proceedings of the 11th Overture Workshop September 2013

Modelling Different CPU Power States in VDM-RT

José Antonio Esparza Isasa and Peter Gorm Larsen

Department of Engineering, Aarhus University

Abstract. With the increasing proliferation of battery-powered embedded devices the need to find the most power efficient way of controlling the micro-controllers inside also increases. In this paper we demonstrate how it is possible to model and analyse the energy consumption of a VDM-RT model. This is done by enabling the CPUs to change between different energy-saving modes and then performing post-analysis of the logged traces and using the data sheets for the processor(s) used to predict the optimal usage of the selected hardware. We also indicate potential future work enhancing the Overture tool to supporting efficiently this kind of analysis.

1 Introduction

In many devices the minimal usage of power from batteries is a competitive parameter that influences the likelihood of consumers selecting your devices. Changing batteries is annoying for the user so maximizing the time between the need for this is advantageous. As a system architect in the early phases the size of the possible design space is enormous. This paper proposes the use of the Vienna Development Method – Real-Time (VDM-RT) dialect [8] and the tool support from the Overture platform [4] to assist the system architect in navigating the design space in pursuit of optimal energy consumption strategies.

The motivation for this work is an Ambient Assistant Living research project called e-stockings[1] where the aim is to produce a device which will assist elderly people with chronical health problems in the use of compression therapy. This device will be controlled by a battery-powered embedded system controlling mechanical parts [3]. The candidate micro-controllers are all able to operate in different power-saving modes, but the optimal use of these features depends on both the properties of the power-saving features of the processors and the way the deployed software utilizes these power-saving features. We believe that it is possible to make use of VDM-RT to give good direction to system architects that have this kind of challenge. In this paper we will present our initial ideas for doing so and focus on a design pattern that we think will be generally applicable for this kind of situation.

This paper proceeds with a short explanation about the specialities of the VDM-RT dialect in Section 2. Then Section 3 explains the concept of power-saving modes in micro-controllers. The main contribution of this paper come in Section 4 where the suggested design pattern is presented. After this different future work areas are presented in

[1] See `http://www.aal-europe.eu/projects/e-stocking/` for more information.

Section 5. Finally Sections 6 and 7 provides references to related work and concluding remarks respectively.

2 The VDM Real-Time Dialect

In this section we will provide a short explanation about the VDM-RT dialect, assuming that the reader already in general is familiar with basic VDM [2, 1]. VDM-RT is one of three dialects supported by the Overture platform including interpretation support [5] that will be exploited in this paper.

The VDM-RT extension is object-oriented and it supports concurrent models. It provides the necessary constructs to represent active classes and incorporates concurrency safety mechanisms. VDM-RT includes the notion of time: a system clock is running from beginning till the end of interpretation. The maximum precision allowed in the interpreter is 1 nanosecond. VDM-RT also provides the notion of processing units in the form of built-in `CPU` classes can be used to declare processing unit and speed (in Hz); different parts of the model are deployed to specified CPUs. CPUs can communicate between themselves through buses. VDM-RT constructs take time to be interpreted, this time is shorter or longer depending on CPU speed. There is also a special kind of CPU, which is present in all the VDM-RT models implicitly, the virtual CPU which per default is infinitely fast and its execution does not affect system timing.

3 CPU Power Modes

CPUs can incorporate different power modes to reduce power consumption. These modes achieve a reduction in power consumption by disabling CPU peripherals and/or reducing the CPU performance. Here we take as an example the ARM Cortex M3 processor in the PSoC5 platform. This CPU has three different power modes available: *Active*, *Sleep* and *Hibernate*. This platform presents several operational modes with different analog and digital resources available and can react to different wake up events on each mode. This feature affects power consumption.

The *active* mode permits code execution and the usage of all platform features, and is the most power consuming state. The *sleep* mode presents a power consumption three orders of magnitude lower than the active consumption and prohibits code execution. Only the bus interface is available. It is possible to switch back to *active* mode after receiving events. Finally, it is possible to enter a *hibernate* mode with the lowest power consumption without turning off the device. This mode does not permit code execution and disables all the system resources. It is possible to transit back to *active* mode after receiving a high level input in the system IO.

4 A Design Pattern to Model CPU Power Modes

The VDM-RT CPUs are constantly able to compute and communicate and thus represent a CPU that is active constantly. This abstraction is not appropriate if we are

interested in studying CPU power consumption. We propose the application of a pattern that makes it possible to represent different CPU power states. This pattern makes use of a Virtual CPU state class to represent different CPU operational modes that controls whether code can be executed or not. An overview of this pattern is shown in the UML class diagram in Fig. 1. The general idea in this pattern is that whenever a thread running application logic is scheduled-in by the VDM-RT scheduler, it will check if the state registered in the Virtual CPU state is active or sleeping before running. There is a second case in which the thread will always run but forces a change in the Virtual CPU state before doing so. Additional details will be provided in the subsections below.
This pattern makes use of the following entities:

`VirtualCPUstate`: represents the CPU state. This class is protected against race conditions and thread interference by mutexes and history counters.
`PowerLogger`: keeps track of the CPU state changes and logs them so they can be analyzed further and represented in a graph.
`ProcTRunner`: represents functionality that should be implemented as a procedural thread.
`RealTimeRunner`: represents functionality that is executed periodically in a periodic thread.

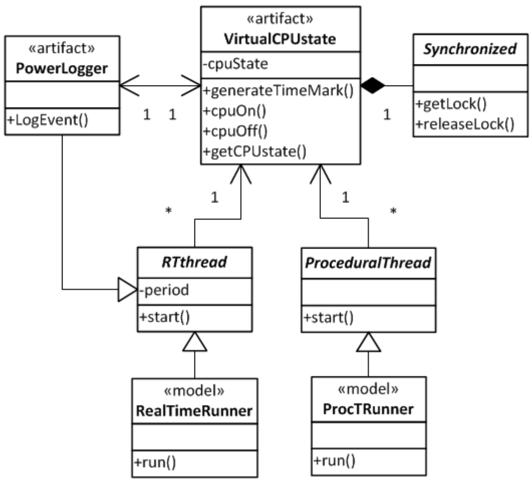


Fig. 1. Design pattern to model multiple CPU states.

The `VirtualCPUstate` class might be accessed by several threads in a concurrent manner. Therefore it is necessary to ensure thread safety in the operations that read and modify the cpuState atrribute that represents the CPU operational mode. We have used mutexes to prevent data corruption.

Additionally and besides preventing the corruption of cpuState, we must ensure that the getCPU state function is executed every time there is a change in the power model. We have reinforced this policy by using permission predicates and history counters. This is shown in listing 1.

```
per turnOn =>  #req(turnOn) - #act(turnOn) = 1;
per turnOff => #req(turnOff) - #act(turnOff) = 1;
per getCPUstate => #active(getCPUstate) = 0  and
    cpuOn => #fin(turnOff) = #fin(getCPUstate);
```

Listing 1: Protection in VirtualState with history counters.

Whenever there is a change in the state of the Virtual CPU a method pushes this change to the PowerLogger class. The operation responsible for this is preceded by a **duration(0)** statement to avoid interfering the application logic timing.

This pattern sets a basic infrastructure to model upon different behaviour. We have modelled two different scenarios that are explained in the subsections below.

4.1 Modelling functionality that makes a CPU become active

In this case we model a scenario in which whenever certain functionality has to be executed the CPU is awakened. This is modelled in the run method modelled in the thread. This is modelled as shown in listing 2: the thread marks the state in the VirtualCPUstate class as on, executes the application logic and marks the CPU state as off again. The state changes are preceded by a 0 nanosecond duration statement, hence we are not accounting for the transition time between states.

```
duration (0) state.turnOn();
executeLogic();
duration (0) state.turnOff();
```

Listing 2: Conditional controlling the execution.

4.2 Modelling functionality that runs if CPU is active

In this second scenario we model the situation in which the application logic runs only when the CPU is active. We have used the structure presented in listing 3. This structure is modelled as part of the thread operation run that is executed when it is scheduled-in. In this case when the thread is thread is scheduled-in it will check the state of the Virtual CPU and determine if it is able to execute its application logic or not.

```
if not state.getCPUstate()
then duration (0) IO`print("\nCPU is off");
else executeLogic();
```

Listing 3: Conditional controlling the execution.

5 Further Work

The design pattern presented in this paper is a way to overcome at the modelling level current limitations of the Overture platform. We would like to take this idea further and possibly incorporate changes in the Overture tool and the VDM-RT language.

5.1 Unaddressed issues

This design pattern does not take into account the communication aspect when the CPU is sleeping. Here we should consider two scenarios: a) Information is received at the communication interface and this generates an interrupt that wakes up the CPU so the information can be processed and b) Information is received and discarded since the CPU is sleeping. These aspects have not been studied but are worthwhile exploring further in later work.

In previous work we studied the possibility of studying power consumption in mechatronic systems by applying co-simulation [3]. The co-simulation approach should be revisited to review its applicability to study computation and communication power consumption. Our initial thesis is that this could be especially beneficial for systems with multiple components (such a SoC with analog and digital blocks). However additional work is needed to confirm this.

5.2 Tool and language modifications

We propose to implement the sleep functionality in the VDM-RT java engine rather than at the modelling level. Additionally we would incorporate a function to register the events that can wake up a CPU from the sleeping state.

```
cpu.wakeOn(event); -- Configure event to wake up CPU
cpu.sleep(); -- Sleep the CPU at some point
```

Listing 4: CPU sleep operations.

Events that could wake up the CPU should be provided by an external periodic thread. It should be possible to associate concrete CPUs to certain events and feed the events to the CPU at a certain point of time.

```
duration (0) if time = eventTime
             then eventGenerator.feed(cpu,event);
```

Listing 5: CPU sleep operations.

Besides these language modifications, additional tool work will be needed to generate consumption graphs automatically and to produce real-time logs in a more effective way.

6 Related Work

Modelling has been widely applied to study energy consumption in previous work but typically at a lower level of abstraction and by using platform specific models. The advantage of this approach is that the more detailed models are more accurate and easier to transition to a final implementation. Additionally the fact that the models are targeting specific platforms improves accuracy as well [7, 6]. Vijaykrishnan et al. make use of modelling in a joint hardware-software approach to optimize energy consumption and using virtual CPUs in [9]. In this case the authors use an ad-hoc modelling platform rather than a generic software modelling language like VDM-RT.

7 Concluding Remarks

This paper has presented a design pattern that can be used for incorporating power consumption considerations to a VDM-RT model. We believe that this can be beneficial for system architects exploring potential strategies for controlling micro-processors in different power saving modes. Naturally this is early work since no special tool support has been incorporated in Overture for supporting these ideas yet. However we believe that it would be possible to develop this so that the applicability for the Overture platform can be increased to also take power consumption considerations into account when desired.

References

1. Fitzgerald, J., Larsen, P.G.: Modelling Systems – Practical Tools and Techniques in Software Development. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, Second edn. (2009), ISBN 0-521-62348-0
2. Fitzgerald, J., Larsen, P.G., Mukherjee, P., Plat, N., Verhoef, M.: Validated Designs for Object–oriented Systems. Springer, New York (2005), `http://www.vdmbook.com`
3. Isasa, J.A.E., Hansen, F.O., Larsen, P.G.: Embedded systems energy consumption analysis through co-modelling and simulation. In: International Conference on Modeling and Simulation, ICMS 2013. World Academy of Science, Engineering and Technology (June 2013)
4. Larsen, P.G., Battle, N., Ferreira, M., Fitzgerald, J., Lausdahl, K., Verhoef, M.: The Overture Initiative – Integrating Tools for VDM. SIGSOFT Softw. Eng. Notes 35(1), 1–6 (January 2010), `http://doi.acm.org/10.1145/1668862.1668864`
5. Lausdahl, K., Larsen, P.G., Battle, N.: A Deterministic Interpreter Simulating A Distributed real time system using VDM. In: Qin, S., Qiu, Z. (eds.) Proceedings of the 13th international conference on Formal methods and software engineering. Lecture Notes in Computer Science, vol. 6991, pp. 179–194. Springer-Verlag, Berlin, Heidelberg (October 2011), `http://dl.acm.org/citation.cfm?id=2075089.2075107`, ISBN 978-3-642-24558-9
6. Mostafa E.A. Ibrahim and Markus Rupp and Hossam A. H. Fahmy: A Precise High-Level Power Consumption Model for Embedded Systems Software. EURASIP Journal on Embedded Systems Volume 2011(1) (January 2011)
7. Sheayun Lee and Andreas Ermedahl and Sang Lyul Min: An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors (2001)
8. Verhoef, M., Larsen, P.G., Hooman, J.: Modeling and Validating Distributed Embedded Real-Time Systems with VDM++. In: Misra, J., Nipkow, T., Sekerinski, E. (eds.) FM 2006: Formal Methods. pp. 147–162. Lecture Notes in Computer Science 4085, Springer-Verlag (2006)

9. Vijaykrishnan, N., Kandemir, M., Irwin, M.J., Kim, H.S., Ye, W.: Energy-driven integrated hardware-software optimizations using simplepower. SIGARCH Comput. Archit. News 28(2), 95–106 (May 2000)

12

Hardware In the Loop for VDM-Real Time Modelling of Embedded Systems

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P51] José Antonio Esparza Isasa, Peter W.V. Jørgensen and Peter Gorm Larsen *Hardware In the Loop for VDM-Real Time Modelling of Embedded Systems*. 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2014) January 2014

The content of this chapter has been excluded due to copyright restrictions, but can be obtained though the respective publisher.

13

ICT-Enabled Medical Compression Stocking for Treatment of Leg-Venous Insufficiency

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P57] Troels Fedder Jensen and Finn Overgaard Hansen and José Antonio Esparza et al. *ICT-Enabled Medical Compression Stocking for Treatment of Leg-Venous Insufficiency*. 7th International Conference on Biomedical Electronics and Devices (BIODEVICES 2014), March 2014.

The content of this chapter has been excluded due to copyright restrictions, but can be obtained though the respective publisher.

14

Architecture for Remote Supervision of Mobile Healthcare Systems

The paper presented in this chapter has been accepted as a peer-reviewed conference paper.

[P44] Finn Overgaard Hansen and Troels Fedder Jensen and José Antonio Esparza *Distributed ICT Architecture for Developing, Configuring and Monitoring Mobile Embedded Healthcare Systems.* International Conference on Health Informatics (HEALTHINF 2014), March 2014.

The content of this chapter has been excluded due to copyright restrictions, but can be obtained though the respective publisher.

Bibliography

[1] Sally Adee. Wireless Sensors that Live Forever. *Spectrum, IEEE*, 47(2):14–14, February 2010.

[2] European Space Agency. ESA's "Sleeping Beauty" Wakes up from Deep Space Hibernation. `http://www.esa.int/Our_Activities/Space_Science/Rosetta/ESA_s_sleeping_beauty_wakes_up_from_deep_space_hibernation`, January 2014. Last access May 2015.

[3] Ahmet Akbas and Yavuz Selim Kalkan. *Handbook of Research on Developments in E-Health and Telemedicine: Technological and Social Perspectives*, volume 1, chapter 8 Designing a Microcontroller-Based Portable MMC/SD Card Recorder: Time and Frequency Domain Analysis of HRV Using Sequential Interbeat Times, pages 170–193. IGI Global, 2010.

[4] Urs Anliker, Jan Beutel, and Matthias Dyer et al. A Systematic Approach to the Design of Distributed Wearable Systems. *IEEE Transactions on Computers*, 53(8):1017 – 1033, August 2004.

[5] M. Ashraf and N. Masoumi. A Thermal Energy Harvesting Power Supply With an Internal Startup Circuit for Pacemakers. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, PP(99), 2015.

[6] Banerjee, A. and Venkatasubramanian, K.K. and Mukherjee, T. and Gupta, S. K S. Ensuring Safety, Security, and Sustainability of Mission-Critical Cyber-Physical Systems. *Proceedings of the IEEE*, 100(1):283–299, 2012.

[7] K. Baynes, C. Collins, E. Fiterman, B. Ganesh, P. Kohout, C. Smit, T. Zhang, and B. Jacob. The Performance and Energy Consumption of Embedded Real-Time Operating Systems. *Computers, IEEE Transactions on*, 52(11):1454–1469, Nov 2003.

[8] Stephen Beeby and Neil White, editors. *Energy Harvesting for Autonomous System*. Smart Materials, Structures, and Systems. Artech House, 2010.

[9] David C. Black, Jack Donovan, Bill Buntion, and Anna Keist. *SystemC: From The Ground Up*. Springer, second edition, 2010.

[10] B. Boehm. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5):61–72, 1988.

[11] J. F. Broenink, P. G. Larsen, M. Verhoef, C. Kleijn, D. Jovanovic, K. Pierce, and Wouters F. Design Support and Tooling for Dependable Embedded Control Software. In *Proceedings of Serene 2010 International Workshop on Software Engineering for Resilient Systems*, pages 77–82. ACM, April 2010.

[12] J. F. Broenink and Y. Ni. Model-Driven Robot-Software Design using Integrated Models and Co-Simulation. In J. McAllister and S. Bhattacharyya, editors, *Proceedings of SAMOS XII*, pages 339 – 344, jul 2012.

[13] B.H. Calhoun, J. Lach, J. Stankovic, D.D. Wentzloff, K. Whitehouse, A.T. Barth, J.K. Brown, Qiang Li, Seunghyun Oh, N.E. Roberts, and Yanqing Zhang. Body Sensor Networks: A Holistic Approach From Silicon to Users. *Proceedings of the IEEE*, 100(1):91–106, January 2012.

[14] Marie Chan, Daniel Esteve, Jean-Yves Fourniols, Christophe Escriba, and Eric Campo. Smart Wearable Systems: Current Status and Future Challenges. pages 137 – 156, September 2012.

[15] MiXiM Community. `http://mixim.sourceforge.net`, January 2015. MiXiM official website.

[16] OMNeT Community. `http://www.omnetpp.org/`, January 2015. OMNeT++ official website.

[17] Controllab products. `http://www.20sim.com/`, January 2013. 20-Sim official website.

[18] Mike Cromheecke. Right-Sizing Your Motion Control System Can Improve Energy-Efficiency. Whitepaper MOTION-WP004A-EN-P, Rockwell Automation, November 2012.

[19] Cypress. PSoC3 and PSoC 5LP Low-Power Modes and Power Reduction Techniques, December 2013.

[20] Brian C. Deese and John P. Holdren. Science and Technology Priorities for the FY 2016 Budget. Memorandum for the Heads of Executive Departments and Agencies 14, Executive Office of the President of the United States, July 2014.

[21] Kevin A Delin and Shannon P. Jackson. The Sensor Web: A New Instrument Concept. In *Proceedings of the SPIE Symposium on Integrated Optics*. Jet Propulsion Laboratory, California Institute of Technology, January 2001.

[22] A. Didioui, C. Bernier, D. Morche, and O. Sentieys. HarvWSNet: A Co-Simulation Framework for Energy Harvesting Wireless Sensor Networks. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 808–812, Jan 2013.

[23] 2015 Multi Annual Strategic Research and Innovation Agenda for the ECSEL Joint Undertaking. Technical report, ECSEL, 2015.

[24] Kerstin Eder, Steven Kerrison, John Gallagher, and Pedro Lopez-Garcia. Whole Systems Energy Transparency. *NANOENERGY Letters*, 1(6):67–68, August 2013.

[25] Martin C. Ekstrøm, Marcus Bergblomma, and Maria Linden et. al. A Bluetooth Radio Energy Consumption Model for Low-Duty-Cycle Applications. *IEEE Transactions on Instrumentation and Measurement*, 61(3):609–617, March 2012.

[26] Horizon 2020, A New Generation of Components and Systems. `http://ec.europa.eu/programmes/horizon2020/en/h2020-section/new-generation-components-and-systems`. Last access May 2015.

[27] Horizon 2020, Advanced Computing. `http://ec.europa.eu/programmes/horizon2020/en/h2020-section/advanced-computing`. Last access May 2015.

[28] Horizon 2020 Work Programme 2014-2015 Leadership in Enabling and Industrial Technologies. Information and Communication Technology Calls. Programme call, European Commission, December 2013.

[29] Laura Marie Feeney and Daniel Willkomm. Energy Framework: An Extensible Framework for Simulating Battery Consumption in Wireless Networks. In *Proceedings of the 3rd International OMNeT++ Workshop*, March 2010.

[30] Daquan Feng, Chenzi Jiang, and Gubong Lim et. al. A Survey of Energy-Efficient Wireless Communications. *IEEE Communications Surveys and Tutorials*, 15(1):167–178, Fist Quarter 2013.

[31] J. S. Fitzgerald and P. G. Larsen. Triumphs and Challenges for the Industrial Application of Model-Oriented Formal Methods. In T. Margaria, A. Philippou, and B. Steffen, editors, *Proc. 2nd Intl. Symp. on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2007)*, 2007. Also Technical Report CS-TR-999, School of Computing Science, Newcastle University.

[32] J. S. Fitzgerald, P. G. Larsen, and M. Verhoef. Vienna Development Method. *Wiley Encyclopedia of Computer Science and Engineering*, 2008. edited by Benjamin Wah, John Wiley & Sons, Inc.

[33] John Fitzgerald, Peter Gorm Larsen, and Marcel Verhoef, editors. *Collaborative Design for Embedded Systems – Co-modelling and Co-simulation*. Springer, 2014.

[34] National Science Foundation. Cyber-Physical Systems Program Solicitation. Technical Report NSF15-541, NSF.

[35] P. Fritzson. Modelica - A Cyber-Physical Modeling Language and the OpenModelica Environment. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 1648–1653, July 2011.

[36] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, January 2004.

[37] Daniel D. Gajski, Samar Abdi, Andreas Gerstlauer, and Gunar Schirner. *Embedded System Design, Modelling Synthesis and Verification*. Springer, 2009.

[38] Junhe Gan, Flavius Gruian, Paul Pop, and Jan Madsen. Energy/Reliability Trade-offs in Fault-tolerant Event-triggered Distributed Embedded Systems. In *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, ASPDAC '11, pages 731–736, Piscataway, NJ, USA, 2011. IEEE Press.

[39] Jack Ganssle, editor. *Embedded Systems: World Class Designs*. Newness World Class Designs. Newness, December 2007.

[40] Editor: Laila Gide. Embedded / Cyber-Physical Systems ARTEMIS Major Challenges: 2014-2020, 2013 DRAFT Addendum to the ARTEMIS-SRA 2011. Technical report, ARTEMISIA, http://www.artemis-ju.eu/publications, december 2013.

[41] Patrick Girard, Nicola Nicolici, and Xiaoqing Wen, editors. *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer, 2010.

[42] Neville Grech, Kyriakos Georgiou, James Pallister, Steve Kerrison, Jeremy Morse, and Kerstin Eder. Static Analysis of Energy Consumption for LLVM IR Programs. In *Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems*, SCOPES '15, pages 12–21, New York, NY, USA, 2015. ACM.

[43] Sandeep K.S. Gupta, Tridib Mukherjee, Georgios Varsamopoulos, and Ayan Banerjee. Research Directions in Energy-Sustainable Cyber–Physical Systems. *Sustainable Computing: Informatics and Systems*, 1(1):57 – 74, 2011.

[P44] Finn Overgaard Hansen, Troels Fedder Jensen, and José Antonio Esparza. Distributed ICT Architecture for Developing, Configuring and Monitoring Mobile Embedded Healthcare Systems. In *International Conference on Health Informatics (HEALTHINF 2014)*, March 2014.

[45] Hans-Peter Hoffmann. Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering. Deskbook Release 4.1, IMB Software Group, 2014.

[46] INCOSE. Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities. Technical report, International Council on Systems Engineering, 7670 Opportunity Rd., Suite 220 San Diego, CA, January 2010.

[47] Texas Instruments. Load Switches: What Are They, Why Do You Need Them And How Do You Choose The Right One? Application note, April 2014.

[48] Texas Instruments. TPS2291XX Load Switch. Datasheet, September 2014.

[P49] José Antonio Esparza Isasa, Finn Overgaard Hansen, and Peter Gorm Larsen. Embedded Systems Energy Consumption Analysis Through Co-Modelling and Simulation. In *Proceedings of the International Conference on Modeling and Simulation, ICMS 2013*. World Academy of Science, Engineering and Technology, June 2013.

[P50] José Antonio Esparza Isasa, Peter W.V. Jørgensen, and Claus Ballegaard. Modelling Energy Consumption in Embedded Systems with VDM-RT. In *Proceedings of the 4th International ABZ conference.*, July 2014.

[P51] José Antonio Esparza Isasa, Peter W.V. Jørgensen, and Peter Gorm Larsen. Hardware In the Loop for VDM-Real Time Modelling of Embedded Systems. In *MODELSWARD 2014, Second International Conference on Model-Driven Engineering and Software Development*, January 2014.

[P52] José Antonio Esparza Isasa and Peter Gorm Larsen. Modelling Different CPU Power States in VDM-RT. In *Proceedings of the 11th Overture Workshop 2013*, Aarhus University, June 2013.

[P53] José Antonio Esparza Isasa, Peter Gorm Larsen, and Kim Bjerge. Supporting the Partitioning Process in Hardware/Software Co-design with VDM-RT. In *Proceedings of the 10th Overture Workshop 2012*, School of Computing Science, Newcastle University, 2012.

[P54] José Antonio Esparza Isasa, Peter Gorm Larsen, and Finn Overgaard Hansen. A Holistic Approach to Energy-Aware Design of Cyber-Physical Systems. *International Journal of Embedded Systems*. Submitted April 2015.

[P55] José Antonio Esparza Isasa, Peter Gorm Larsen, and Finn Overgaard Hansen. Energy-Aware Design of Embedded Software through Modelling and Simulation. In *Proceedings of the SYRCoSE 2014 symposium*, May 2014.

[P56] José Antonio Esparza Isasa, Peter Gorm Larsen, and Finn Overgaard Hansen. Energy-Aware Model-Driven Development of a Wearable Health Care Device. In *Presented in 4th Symposium FHIES/SEHC, In Press*, June 2014.

[P57] Troels Fedder Jensen, Finn Overgaard Hansen, and José Antonio Esparza et al. ICT-Enabled Medical Compression Stocking for Treatment of Leg-Venous Insufficiency. In *International Conference on Biomedical Electronics and Devices (BIODEVICES 2014)*, March 2014.

[58] Peter W.V. Jørgensen, Luís D. Couto, and Morten Larsen. A Code Generation Platform for VDM. In *The Overture 2014 workshop*, June 2014.

[59] Sandeep Kamath. Measuring Bluetooth Low Energy Power Consumption. Application note, Texas Instruments, October 2010.

[60] M Amin Karami and Daniel J Inman. Powering Pacemakers from Heartbeat Vibrations Using Linear and Nonlinear Energy Harvesters. *Applied Physics Letters*, 100(4):042901, 2012.

[61] Steve Kerrison and Kerstin Eder. Energy Modelling of Software for a Hardware Multithreaded Embedded Microprocessor. *ACM Transations on Embedded Computing Series*, 14(3), April 2015.

[62] Kyoung-Dae Kim and P.R. Kumar. Cyber-Physical Systems: A Perspective at the Centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, 2012.

[63] C. Kleijn. *20-sim 4.1 Reference Manual*. Controllab Products B.V., Enschede, First edition, 2009. ISBN 978-90-79499-05-2.

[64] C. Kleijn. *20-sim 4C 2.1 Reference Manual*. Controllab Products B.V., Enschede, First edition, 2013. ISBN 978-90-79499-05-2.

[65] Costis Kompis and Prateek Sureka. Power Management Technologies to Enable Remote and Wireless Sensing. Technical Report, Knowledge Transfer Network. Sensors and Instrumentation, May 2010.

[66] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis, and T. Laopoulos. Energy Consumption Estimation in Embedded Systems. *Instrumentation and Measurement, IEEE Transactions on*, 57(4):797–804, April 2008.

[67] Chong-Min Kyung and Sungjoo Yoo, editors. *Energy-Aware System Design. Algorithms and Architectures*. Springer, 2011.

[68] A. Lal, Rajesh Duggirala, and Hui Li. Pervasive Power: A Radioisotope-Powered Piezoelectric Generator. *Pervasive Computing, IEEE*, 4(1):53–61, Jan 2005.

[69] Peter Gorm Larsen, Nick Battle, Miguel Ferreira, John Fitzgerald, Kenneth Lausdahl, and Marcel Verhoef. The Overture Initiative – Integrating Tools for VDM. *SIGSOFT Softw. Eng. Notes*, 35(1):1–6, January 2010.

[70] Peter Gorm Larsen, Sune Wolff, Nick Battle, John Fitzgerald, and Ken Pierce. Development Process of Distributed Embedded Systems using VDM. Technical Report TR-2010-02, The Overture Open Source Initiative, April 2010.

[71] Kenneth Lausdahl, Peter Gorm Larsen, and Nick Battle. A Deterministic Interpreter Simulating A Distributed Real Time System Using VDM. In Shengchao Qin and Zongyan Qiu, editors, *Proceedings of the 13th international conference on Formal methods and software engineering*, volume 6991 of *Lecture Notes in Computer Science*, pages 179–194, Berlin, Heidelberg, October 2011. Springer-Verlag. ISBN 978-3-642-24558-9.

[72] Edward A. Lee. Cyber Physical Systems: Design Challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, Jan 2008.

[73] Edward A. Lee. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors*, (15):4837–4869, February 2015.

[74] Umer Liqat, Steve Kerrison, Alejandro Serrano, Kyriakos Georgiou, Pedro Lopez-Garcia, Neville Grech, Manuel V. Hermenegildo, and Kerstin Eder. Energy Consumption Analysis of Programs Based on XMOS ISA-Level Models. In Gopal Gupta and Ricardo Peña, editors, *Logic-Based Program Synthesis and Transformation*, volume 8901 of *Lecture Notes in Computer Science*, pages 72–90. Springer International Publishing, 2014.

[75] J.R. Lorch and A.J. Smith. Software Strategies for Portable Computer Energy Management. *Personal Communications, IEEE*, 5(3):60–73, Jun 1998.

[76] Stuart Hunt & Associates Ltd. Nickel-63 Radioactive Material Safety Data Sheet. Technical report, May 2015.

[77] Mehdi Maasoumy Haghighi. Controlling Energy-Efficient Buildings in the Context of Smart Grid: A Cyber Physical System Approach. Technical Report UCB/EECS-2013-244, EECS Department, University of California, Berkeley, Dec 2013.

[78] Jens Kargaard Madsen, Henrik Karstoft, Finn Overgaard Hansen, and Thomas Skjødeberg Toftegaard. ASE-BAN, a Wireless Body Area Network Testbed. In *EMERGING 2010 : The Second International Conference on Emerging Network Intelligence*, pages 120–124, Florence, Italy, October 2010. Xpert Publishing Service.

[79] Clive Maxfield. *FPGAs Instant Access*. Newnes, 2008.

[80] M. Nistor, D.E. Lucani, and J. Barros. A Total Energy Approach to Protocol Design in Coded Wireless Sensor Networks. In *Network Coding (NetCod), 2012 International Symposium on*, pages 31–36, 2012.

[81] The Institute of Electrical and Inc. Electronics Engingeers. 1364-2005 - IEEE Standard for Verilog Hardware Description Language. Technical report, IEEE, NY, USA, 2006.

[82] The Institute of Electrical and Inc. Electronics Engingeers. IEEE Standard VHDL Language Reference Manual. Technical report, IEEE, NY, USA, January 2009.

[83] The Institute of Electrical and Inc. Electronics Engingeers. 1666-2011 - IEEE Standard for Standard SystemC Language Reference Manual. Technical report, IEEE, NY, USA, January 2012.

[84] Ole-Jørgen Ohm and Dejan Danilovic. Improvements in Pacemaker Energy Consumption and Functional Capability: Four Decades of Progress. *Pacing and Clinical Electrophysiology (PACE)*, 20(1):2–9, January 1997.

[85] OMG. OMG Unified Modeling Language (OMG UML) Superstructure ver. 2.4.1. *www.uml.org/spec/UML/2.4.1/Superstructure/PDF/*, 2011.

[86] Bassem Ouni, Cecile Belleudy, and Eric Senn. Accurate Energy Characterization of OS Services in Embedded Systems. *EURASIP Journal on Embedded Systems*, 2012(1):6, 2012.

[87] Ozgur Celebican and Tajana Simunic Rosing and Vincent J. III Mooney. Energy Estimation of Peripheral Devices in Embedded Systems. In *Proceedings of the 14th ACM Great Lakes symposium on VLSI*, GLSVLSI '04, 2004.

[88] Alexandros Pantelopoulos and Nikolaos Bourbakis. SPN-Model based Simulation of a Wearable Health Monitoring System. In *Proceedings of the 31st Annual International Conference of the IEEE EMBS*, pages 320–323. IEEE, September 2009.

[89] Chulsung Park, K. Lahiri, and A. Raghunathan. Battery Discharge Characteristics of Wireless Sensor Nodes: an Experimental Analysis. In *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, pages 430–440, 2005.

[90] Young-Hwan Park, Sudeep Pasricha, Fadi J. Kurdahi, and Nikil Dutt. A Multi-Granularity Power Modeling Methodology for Embedded Processors. *IEEE Trans. Very Large Scale Integr. Syst.*, 19(4):668–681, 2011.

[91] L. Parolini, B. Sinopoli, B.H. Krogh, and Zhikui Wang. A Cyber Physical Systems Approach to Data Center Modeling and Control for Energy Efficiency. *Proceedings of the IEEE*, 100(1):254–268, Jan 2012.

[92] J Paulo and PD Gaspar. Review and Future Trend of Energy Harvesting Methods for Portable Medical Devices. In *Proceedings of the world congress on engineering*, volume 2, 2010.

[93] Birgit Penzenstadler and Jonas Eckhardt. A Requirements Engineering Content Model for Cyber-Physical Systems. In *RESS*, pages 20–29, 2012.

[94] Douglas L. Perry. *VHDL: Programming by Example*. McGraw Hill, fourth edition, 2002.

[95] Gianfranco Pistoia. *Battery Operated Devices and Systems. From Electronics to Industrial*. Elsevier, 2009.

[96] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-Physical Systems: The Next Computing Revolution. In *Proceedings of the Design Automation Conference, Ahaheim, California 2010*. ACM, 2010.

[97] Ravishankar Rao, Sarma Vrudhula, and Daler N. Rakhmatov. Battery Modeling for Energy-Aware System Design. *Computer*, 36(12):77–87, December 2003.

[98] Augusto Ribeiro, Kenneth Lausdahl, and Peter Gorm Larsen. Run-Time Validation of Timing Constraints for VDM-RT Models. In Sune Wolff and John Fitzgerald, editors, *Proceedings of the 9th Overture Workshop*, number ECE-TT-2 in Technical Report Series, pages 4–16, June 2011.

[99] Rick Steiner Sandford Friedenthal, Alan Moore. *A Practical Guide to SysML*. Morgan Kaufman OMG Press, Friendenthal, Sanford, First edition, 2008. ISBN 978-0-12-374379-4.

[100] Findlay Shearer. *Portable Electronics*, chapter System-Level Approach to Energy Conservation, pages 85–128. World Class Designs. Newness, 209.

[101] Sheayun Lee and Andreas Ermedahl and Sang Lyul Min. An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors, 2001.

[102] A. Sinha and A. Chandrakasan. Dynamic Power Management in Wireless Sensor Networks. *Design Test of Computers, IEEE*, 18(2):62–74, Mar 2001.

[103] Mircea R. Stan and Kevin Skadron. Guest Editors' Introduction: Power-Aware Computing. *Computer*, 36(12):35–38, December 2003.

[104] OMG Systems Modeling Language (OMG SysML$^{\mathrm{TM}}$). Technical Report Version 1.2, SysML Modelling team, June 2010. http://www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf.

[105] Janos Sztipanovits and Susan Ying. Foundations for Innovation: Strategic R&D Opportunities for 21st Century Cyber-Physical Systems. Technical report, Steering Committee for Foundations for Innovation in Cyber-Physical Systems, January 2013.

[106] Andrea Tundis, Lena Rogovchenko-Buffoni, Peter Fritzson, and Alfredo Garro. Modeling System Requirements in Modelica: Definition and Comparison of Candidate Approaches. In *5th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, University of Nottingham, UK, April 2013.

[107] Unsal, O.S. and Koren, I. System-Level Power-aware Design Techniques in Real-Time Systems. *Proceedings of the IEEE*, 91(7):1055–1069, 2003.

[108] Job van Amerongen. *Dynamical Systems for Creative Technology*. Controllab Products, Enschede, Netherlands, 2010.

[109] Timo Vuorela. *Technologies for Wearable and Portable Physiological Measurement Devices*. PhD thesis, Tampere University of Technology, 2011.

[110] A.Y. Wang and C.G. Sodini. A Simple Energy Model for Wireless Microsensor Transceivers. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 5, pages 3205–3209 Vol.5, 2004.

[111] Qin Wang, M. Hempstead, and W. Yang. A Realistic Power Consumption Model for Wireless Sensor Network Devices. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society*, volume 1, pages 286 –295, sept. 2006.

[112] Tim Wescott. PID Without a PhD. *Embedded Systems Design*, (86 - 108), October 2000.

[113] Fumin Zhang, Zhenwu Shi, and Wayne Wolf. A Dynamic Battery Model for Co-design in Cyber-Physical Systems. In *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, ICDCSW '09, pages 51–56, Washington, DC, USA, 2009. IEEE Computer Society.

[114] Bo Zhou, A. Dvoryanchikova, A. Lobov, and J.L. Martinez Lastra. Modeling System of Systems: A Generic Method Based on System Characteristics and Interfaces. In *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, pages 361 –368, july 2011.