World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:7, No:6, 2013

Embedded Systems Energy Consumption Analysis Through Co-modelling and Simulation

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

Open Science Index, Computer and Systems Engineering Vol:7, No:6, 2013 publications.waset.org/12904.pdf

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).

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

[2] DESTECS official website: http://www.destecs.org/

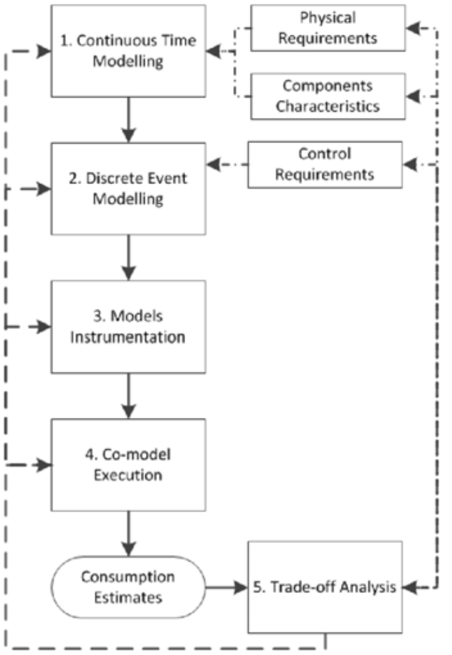
1. Continuous Time Modelling
Physical Requirements
Components Characteristics
2. Discrete Event Modelling
Control Requirements
3. Models Instrumentation
4. Co-model Execution
Consumption Estimates
5. Trade-off Analysis

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).

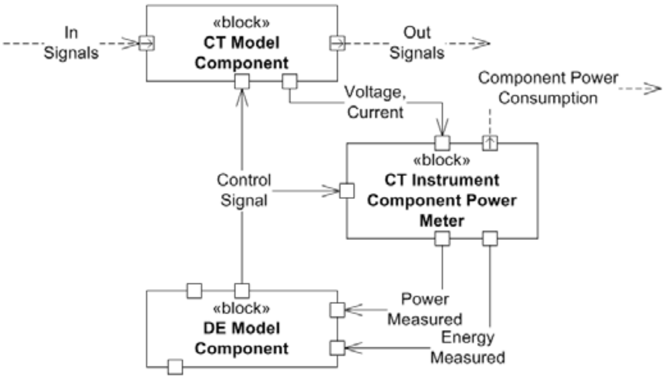
In Signals
«block» CT Model Component
Out Signals
Voltage, Current
Component Power Consumption
Control Signal
«block» CT Instrument Component Power Meter
«block» DE Model Component
Power Measured
Energy Measured

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.

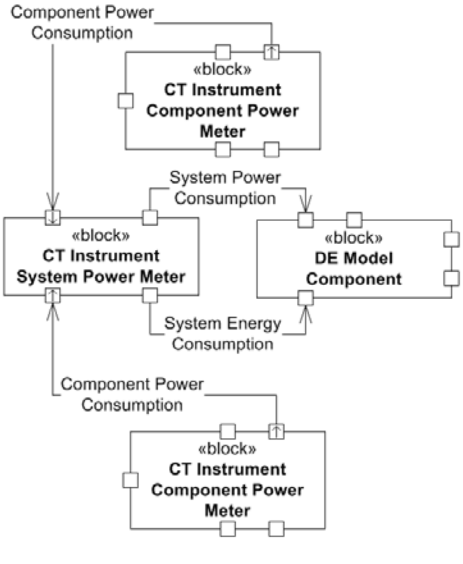
Component Power Consumption
«block» CT Instrument Component Power Meter
System Power Consumption
«block» CT Instrument System Power Meter
«block» DE Model Component
System Energy Consumption
Component Power Consumption
«block» CT Instrument Component Power Meter

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.

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.

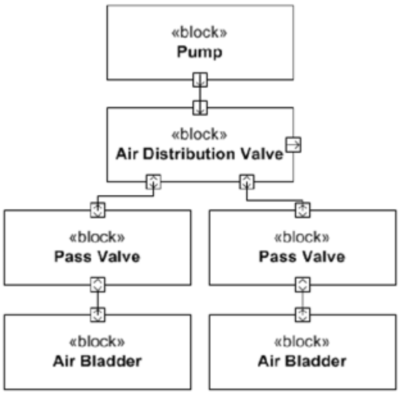

Fig. 4. SysML block diagram of the pneumatic subsystem.

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.

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

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.

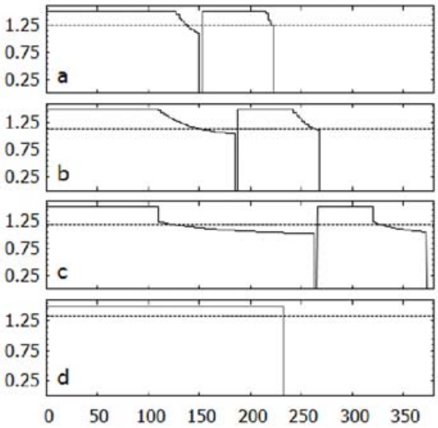

Fig. 5. System power consumption over time under different control algorithms for stocking compression. y axis: power in watts, x axis: time in seconds.

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.

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.