A Methodology for Designing Energy-aware Secure Embedded Systems

Mehrdad Saadatmand, Antonio Cicchetti, Mikael Sjödin
Mälardalen Real-Time Research Centre (MRTC)
Mälardalen University, Västerås, Sweden
{mehrdad.saadatmand, antonio.cicchetti, mikael.sjodin}@mdh.se

***Abstract*—Bringing security aspects in earlier phases of development is one of the major shifts in software development trend. Model-driven development which helps with raising the abstraction level and facilitating earlier analysis and verification is a promising approach in this regard and there have been several efforts on modeling security aspects. However, the issue is that when it comes to embedded systems, non-functional requirements such as security are so interconnected that in order to satisfy one, trade-off analysis with other ones are necessary. Energy consumption is one of these requirements which is of great importance in embedded systems domain due to resource limitations that these systems have. In this paper, focusing on security and energy consumptions we propose a new methodology for model-driven design of embedded systems to bring energy measurements and estimations earlier in development phases and thus identify security design decisions that cause violations of specified energy requirements.**

***Index Terms*—Secure Embedded Systems, Energy Consumption, Modeling, MDA.**

I. Introduction

In the design of embedded systems, security is usually not considered as a high-level design consideration, especially that it has been mainly used to refer to physical security and physical access protection [1]. However, the rapidly growing use of embedded systems in critical applications (e.g. medical devices), and the increase in the use of interconnected and communicating embedded devices, make security consideration more important that ever in the design process of embedded systems.

As a non-functional requirement, satisfying security has impacts on other requirements in the system such as power requirements, maintainability, and performance. For example, implementing encryption algorithms can incur heavy computational capacity and thus affect the performance of the system. In embedded systems with limited resources, this can easily lead to exceeding and overusing memory and processor capacity and paralyzing other and perhaps highly critical functions of the system. Using CPU and memory for longer periods of time also means consuming more energy and in some systems using battery and affecting system's up/life-time. Therefore, security aspects not only need to be considered at early design phases in the development of embedded systems (and not as add-on to the system design later on), but also there needs to be ways to co-design it with other aspects to facilitate analysis of the implications of security requirements on them [2]. In this paper, we introduce a model-driven methodology to design secure embedded systems considering their energy consumptions. Using a model-driven approach helps to increase the abstraction level and cope with the design complexity of embedded systems. It enables to perform analysis using the design model of the system and therefore identify issues earlier than implementation phase.

Due to the importance of energy aspects in embedded systems, there are efforts on calculating energy values. On the other hand, several solutions for modeling security have been offered such as [3] and [4]. The problem is that these works have usually been done in isolation while these aspects have direct impact on each other in embedded system. What is lacking is a general approach which offers a design methodology so that these separate works can be well adopted in order to provide a roadmap for better design of embedded systems. Such a methodology can serve as the unifying roadmap, help to identify interfaces and assumptions that separate efforts on designing energy-aware secure embedded systems should/can take into account, and provide them directions. This is basically the contribution of our work in this paper.

In order to provide energy values as much as possible, we incorporate both actual measurement (e.g. measurements done using oscilloscope) and estimation approaches for energy values. This is accommodated in the methodology by introducing Energy Consumption Model. In the scope of this paper, we describe the methodology, why it is needed, and different types of models used in it. To implement it, we are currently working on defining a meta-model for the energy consumption model. Obtaining and measuring energy costs are beyond the scope and space of this paper, but we provide pointers to two different works on this topic.

The remainder of the paper is structured as follows. Section II describes the parts that constitute different steps in our methodology. In this section we also describe our vision for extension of the methodology. In section III, we show an example on how it is possible to implement different parts of the methodology. Section IV, points out some related projects and in section V we discuss future work and directions.

II. Modeling Approach

Part A, B, C and D in figure 1 constitute the core body of our suggested approach to bring energy-awareness to designed security features. It follows Model-Driven Architec-

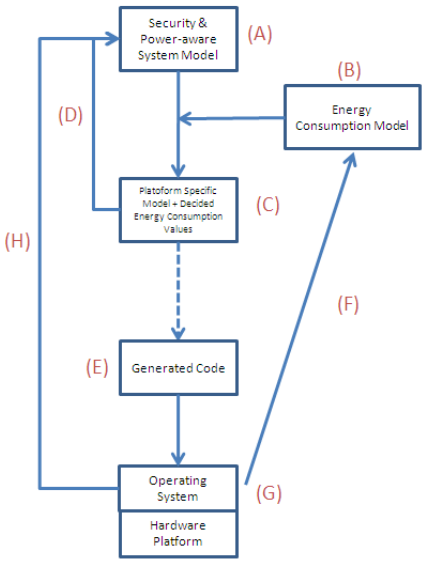

Fig. 1. Complete design methodology vision

ture (MDA) approach of distinguishing Platform Independent Models (PIM) and Platform Specific Models (PSM). Having appropriate modeling formalism, system designer creates a high level model of the system including specified security and energy consumption requirements. Then when it comes to transforming this model to a PSM, an energy consumption model of the platform is used in the transformation to *decide* values for security features defined in the model. We use the term 'decide' here because for some features a precise energy measurement can be provided while for others only an estimation can be done. The generated PSM model is a read-only model with energy values which is used as a basis to validate the design model. Therefore, we do not pollute the design model of the user with detailed energy values and through a feedback loop (D) we mark design decisions that violate the specified requirements. This process is repeatable until a satisfactory design is achieved. By keeping PSM read-only and allowing changes only on the design model, we ensure having only one modifiable source of information and consistency of different generated models. The subtleties and challenges of each step are discussed in the following sections.

A. Design Model

The main challenge in the design model is how to specify non-functional requirements and in particular security and power requirements and their attributes. To cover these needs, we have proposed a UML profile (in [5]) based on MARTE [6] and adopted concepts from SysML [7] for requirements and UMLsec [4] for security aspects. Specification of requirements and the relation between them and model elements satisfying them are modeled using SysML requirements diagram and its association types for requirements (*Satisfy*, *Refine*, *Copy*, *DeriveReqt* and *Trace*). This is necessary in order to establish traceability between different requirements and also requirements and model elements in our model. Security features are defined by applying UMLsec concepts and energy aspects and definition of model elements' attributes are achieved by adopting MARTE. Also other missing necessary security concepts are compensated for by defining them using MARTE.

B. Energy Consumption Model

In its simplest form, this model keeps measurements and formulas for energy consumption of different platforms. In the scope of this work, only security related items are considered here, however, as we will see it can be easily used to contain energy consumption information for other features of the system. This model can be built using MARTE and especially its Value Specification Language (VSL) package.

[8] is an example of measurement approaches for energy consumption of different security algorithms on two hardware platforms used in Wireless Sensor Networks. On the other hand, [9] offers a model-based approach for calculating energy consumption in OS-based embedded systems. It provides formulas and calculation models for estimation of energy costs such as consumption laws for stand-alone tasks considering different cache-usage scenarios. In order to provide as much information as possible about energy consumption of the designed system at this high level, a combination of both approaches (actual measurements and calculation models and formulas) is used in this model.

C. Platform Specific Model

The platform specific model (since it is generated based on a specific platform energy consumption model) is the first of potentially several platform specific models that are generated in the transformation towards the final code. Annotated with decided energy values for the modeled features, this model is used to find out requirement violations in the design model. The related model elements in the design model are marked for the user in order to apply appropriate modifications. The values in this model can also help with performing trade-off analysis and optimization on the design model. As mentioned before, this model is real-only and the only model which can be modified is the design model. This is a key point in our model-driven approach to have single source of information and modification to keep consistency among generated models and ensure validity of traceability links among their elements.

D. Methodology Vision

Figure 1 as a whole shows the target structure of our suggested methodology. Our vision is to incorporate the above mentioned parts into the methodology that is depicted in this figure. In this model, the role of other system parts such as platform is added to the picture. The platform plays an important role in guaranteeing that the non-functional requirements that have been defined in the design model are actually preserved during runtime. At runtime the operating system performs monitoring on consumption values and with the provided values the running code can adapt itself to stick to the specified energy requirements or report violations back to the system model (link H in the figure). The actual

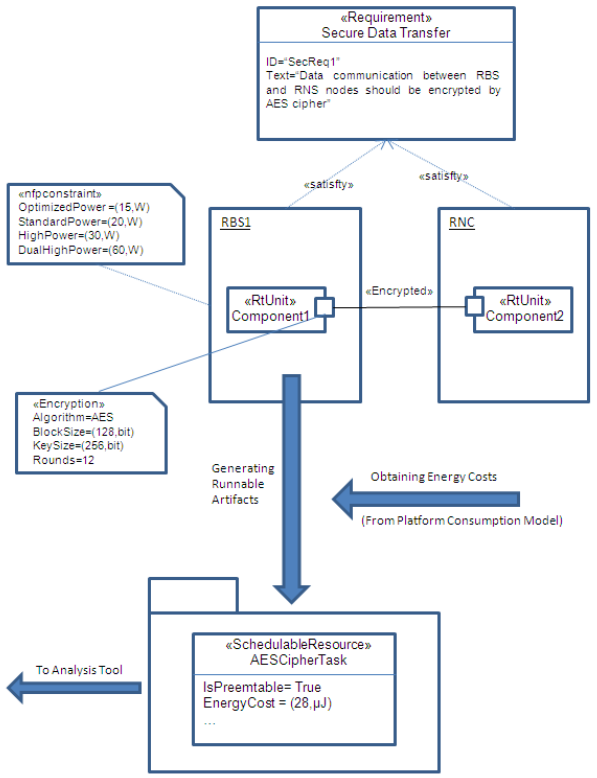

Fig. 2. From design model to runnable artifacts considering energy costs (RBS: Radio Base Station, RNC: Radio Network Controller)

measured values by the platform during runtime can also be used to correct and extend the energy consumption model of the platform (link F). This feedback link has the potential to provide the following interesting feature: A system can be designed with minimal information in the platform energy consumption model. Each execution of the system propagates the energy consumption model with additional measured values and provides more accurate decisions for future systems developed on similar platforms.

III. Implementation Example

In our example case (from telecommunication systems), there are two components running on different nodes with different power specifications. The communication between these two nodes should be encrypted over a secure media. High-level power requirements are defined and linked to the components using SysML Requirements concepts and association types (e.g. *Satisfy*). The link between the two components is marked with Encrypted stereotype from UMLsec and additional properties for the encryption are defined based on MARTE. We have defined an *Encryption* stereotype having *Algorithm*, *BlockSize*, *KeySize*, and *Rounds* attributes using MARTE Non-Functional Properties (NFP) package. This stereotype is used on the output port to specify the details of the encryption which satisfies the security requirement put on the sending node to encrypt all data before transmission.

In the platform consumption model, we have defined a *Platform* class which represents name and specification of a specific platform using MARTE concepts particularly Hardware Resource Modeling (HRM) package. An *EnergyCost* class is also defined which is associated to the platform class and has *ActionCategory*, *Action* and *Cost* attributes. The *ActionCategory* is an enumeration containing "Security" as its only item for the moment (this enumeration is used for extension beyond security). *Action* is to hold a unique and human-readable string used in transformation engine to match with security feature defined in the design model, and *Cost* is the actual measured energy cost for that *Action* or a calculation formula. The variables in the formula are instantiated with values during transformation using the parameters from the design model. The platform-specific model generated from the transformation phase contains the platform-based realization elements for the specified components; e.g. using *SchedulableResource* stereotype of MARTE and decided energy cost on them using MARTE NFP concepts. An analysis tool is to be designed to use this PSM model for validating feasibility of the design model considering its energy implications.

Figure 2 shows part of this implementation (due to space limit) and how the concepts mentioned so far work together to model the system. Security and energy requirements and properties of the system are defined in the system model (using MARTE, SysML and UMLsec). An energy cost is decided using platform consumption model for the generated runnable task which implements Component1. This energy cost along with those of other generated security elements are used in the analysis phase in evaluation against the specified power values of the nodes. The designer can then apply necessary design changes if the energy costs of security features are for instance low, too high or violating maximum allowed levels.

IV. Related Industrial Projects and Works

In this section, we briefly describe some related industrial projects that contribute to implementing our methodology, demonstrating the feasibility of our approach and its relevance and benefits for industry. There are several projects (some still in progress) whose scope and results match different parts of our suggested methodology.

CHESS project [10]: CHESS is an on-going European project which focuses on preservation of non-functional attributes in real-time and dependable embedded systems. It introduces a model-driven engineering approach which provides guarantees for preservation of non-functional requirements through the use of static analysis and runtime monitoring and establishing a feedback mechanism from platform to the design model to mark violations. CHESS results can contribute to parts A, D, G and H of our methodology shown in 1.

GEODES project [11]: This European project provides design techniques for design and global energy optimization in distributed embedded systems. It covers issues like power-aware middlewares and protocols. It is related to part F and G of the methodology.

Modeling non-functional requirements in telecommunication systems [5]: This is a work that we have done with

Ericsson and still extend to provide better modeling solutions on non-functional requirements in telecommunication systems. The part which is already done is a UML profile-based approach adopting concepts from MARTE, SysML and UMLsec for modeling of typical non-functional requirements dominant in telecommunication systems. This work contributes to part A.

Also as mentioned before [9] provides formulas, calculation models and a tool framework for estimation and integration of power values in earlier phases of development in a model-based approach. It provides a solution for high level modeling and estimation of power aspects in embedded systems that rely on operating systems. It can contribute to part B, F and G. The study in [8] is a representative of measurement approaches to decide upon energy costs of security algorithms. It has basically served as a sample and motivation work for part A,B and C.

V. Discussion and Future Works

Adding other aspects to the picture: In this paper,we focused on relation of security and energy aspects in embedded systems. Currently, we are working on defining a meta-model for the energy consumption model to introduce platform information (here energy) in the transformations. On the other hand, the same discussions regarding interconnection and mutual impacts of energy and security hold true for other aspects such as performance, dependability, etc. In our suggested approach, we have based our solution on MARTE which covers a broad range of concepts in real-time embedded systems such as timing and schedulability, performance, allocation and distribution, synchronization mechanisms and facilitates definition of arbitrary non-functional properties. This makes it possible to cover versatile non-functional requirements in the model. While in this work, we started from a model containing both security and energy requirements, an extension for the envisioned methodology could be to have one single model and several views of that model such as security view, energy consumption view, timing and scheduling view and so on. So basically, any modification in one aspect (e.g. timing) is only allowed in its specific view which in the end is actually stored in the model. This approach is being developed in the scope of CHESS project.

UML profiles versus DSL: It is important to mention that there are alternative solutions to implement the suggested methodology. For example, another way of doing that is through the use of Domain Specific Languages (DSL) [12] for different aspects that are used in the methodology: A DSL to model security and one for modeling energy (and probably others for modeling different aspects such as performance). Different models modeled by specific DSLs will then be *woven* together and transformed to generate the code. There are some advantages and disadvantages of using DSLs. By using DSLs for each model, it is possible to extend the definition (language) of each aspect (security or energy) separately without causing modifications for other aspects. Also adding other aspects beyond security and energy such as performance becomes easier (by defining an appropriate DSL for performance). This is especially helpful in trade-off analysis and when an overall higher priority is assigned to one aspect than another. For example, in a system where performance is given higher importance than security, design decisions contributing to high performance can be kept intact while security features can be reduced or *filtered* from using for code-generation to solve energy requirements violations. Also, having separation between aspects using different DSLs facilitates system configuration. It becomes possible to have different security models with different attributes and choose which one to apply (e.g. high, medium or low security configurations). On the other hand, using different DSLs makes the transformation engine a lot more complicated compared to UML profile-based approach. Consistency between different models is also another issue which should be taken care of. Moreover, defining each DSL and weaving models, and learning different modeling languages for developer teams are extra steps which require additional efforts. To read more about different advantages and disadvantages of DSLs versus UML profiles refer to [12] and [13].

References

[1] S. Gürgens, C. Rudolph, A. Maña, and S. Nadjm-Tehrani, "Security engineering for embedded systems: the secfutur vision," in *Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems*, ser. S&D4RCES '10. New York, NY, USA: ACM, 2010, pp. 7:1–7:6. [Online]. Available: http://doi.acm.org/10.1145/1868433.1868443

[2] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Trans. Embed. Comput. Syst.*, vol. 3, pp. 461–491, August 2004. [Online]. Available: http://doi.acm.org/10.1145/1015047.1015049

[3] T. Lodderstedt, D. Basin, and J. Doser, "Secureuml: A uml-based modeling language for model-driven security." Springer, 2002, pp. 426–441.

[4] J. Jürjens, "Umlsec: Extending uml for secure systems development," in *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*. London, UK: Springer-Verlag, 2002, pp. 412–425.

[5] M. Saadatmand, A. Cicchetti, D. Corcoran, and M. Sjödin, "Toward a tailored modeling of non-functional requirements for telecommunication systems," in *Proceedings of 8th International Conference on Information Technology : New Generations*, Las Vegas, Nevada, USA, 2011.

[6] MARTE specification version 1.0 (formal/2009-11-02), http://www.omgmarte.org.

[7] OMG SysML Specifcation V1.2, June 2010, http://www.sysml.org/specs.htm.

[8] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Computer Networks*, June 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.05.011

[9] S. Dhouib, "High level power modelling and estimation for os-based embedded systems," Ph.D. dissertation, University of South Brittany, France, 2009.

[10] "CHESS Project," http://chess-project.ning.com/, December 2010.

[11] "GEODES Project," http://www.itea2.org/public/project_leaflets/GEODES_profile_oct-08.pdf/, December 2010.

[12] I. Weisemöller and A. Schürr, "A comparison of standard compliant ways to define domain specific languages." Berlin, Heidelberg: Springer-Verlag, 2008, pp. 47–58.

[13] B. Selic, "A systematic approach to domain-specific language design using uml," in *ISORC '07: Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 2–9.