A design methodology for systematic Multiple properties-based Partitioning of embedded systems applications
# Table of Contents

Abstract ........................................................................................................................................... 3

1 Research Description .................................................................................................................. 4
   1.1 Problem Formulation ............................................................................................................ 4
   1.2 Main Research Objective and Research Questions ............................................................... 5
   1.3 Research Methodology ........................................................................................................ 6
   1.4 Contribution ....................................................................................................................... 8

2 Thesis Outline ............................................................................................................................ 14

3 Related Work ............................................................................................................................. 16

4 Current Status and Time Plan .................................................................................................. 19
   4.1 Publications ....................................................................................................................... 19
      4.1.1 Publications Fundamental to the Thesis Contributions .................................................. 19
      4.1.2 Publications Related to the Thesis ................................................................................ 19
   4.2 Courses .............................................................................................................................. 20
   4.3 Scheduled Activities and Time Plan ................................................................................... 20

5 References ................................................................................................................................... 22
Abstract

The growing use of heterogeneous platform (e.g. platforms consisting of more computational units e.g. Field-Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPUs) and Multicore Central Processing Units (CPUs)) together with the increases in complexity of embedded applications constantly challenges engineers and researchers in finding new methodologies for the partitioning of such applications in form of executable hardware or software. In this scenario, the partitioning process is complex and cannot be focused only on physical/performance metrics, but it has to be based on decisions accounting high-level application requirements (such as sustainability, security, maintainability) as well as business goals and project constraints (e.g. resources availability and expertise, time-to-market, legacy, etc.). Although, over the past decades, several different partitioning approaches have been proposed, they lack the possibility of providing optimal partitioning solution which decisions are based on a larger number of application requirements as well as business goals and project constraints. In industry it is common practice to carry out the partitioning in an early stage and based on decisions relying on hardware and software team expertise. This lead to a number of side-effects such as redesigns, design flow interruptions, unplanned interactions due to uncontrolled dependencies between hardware and software parts which negatively impact the overall development process in term of quality and costs. To decrease these negative effects we introduce a new approach that aims at designing systems in a technology-independent way, and performing the decisions about partitioning in the later design phase, based on many criteria related to run-time and lifecycle properties, and the business goals and development project constraints. The main objective of this licentiate thesis is the design of a multiple decision criteria-based partitioning and systematic methodology for hardware and software deployment of embedded applications into heterogeneous platforms. The main contribution is summarized as follows: a) The design of the process flow, the set of activities and artifacts enabling the partitioning, b) the specification of metamodels suitable for both capturing the hardware and software aspects of the embedded applications and their related extra-functional properties(EFPs), c) categorization of component EFPs in respect to partitioning and analysis of partitioning impact on EFPs for the Control and Automation domain, d) the suitability assessment of MCDA methods for achieving the partitioning solution and e) finally the identification of a tool chain able of supporting the proposed partitioning process. In order to to validate the proposed methodology applicability, we provide a proof-of-concept on an industrial case study. Specifically, in the proposal we illustrate the research context (problem and research questions), the research approach and our contributions. In addition we present the current status of our research, the scheduled activities and time plan up to the licentiate defense.
1 Research Description

This section will start discussing the research problem and its formalization. Then, the main objectives of the research work and related research questions will be presented. Subsequently, the undertaken research methodology and the methods applied will be described. Finally, a description of the contributions is provided. It will describe the overall contribution in relation to the main objective and the included publications in the licentiate thesis.

1.1 Problem Formulation

More and more industrial embedded system applications need today to be deployed on heterogeneous platforms (i.e. platforms consist of different computation resources, e.g. CPUs, FPGA, and GPU). This is due to the increase in complexity of the applications in terms of features, constraints, and utilization scenarios that they have to provide and support. During the development - specifically at the design phase - it is of crucial importance to decide upon the application partitioning and related deployment, i.e. what functions are designed. Implemented and deployed as software executable units (e.g. compiled C/C++ code, etc.) or which as hardware (e.g. synthesized from VHDL, etc.). These architectural decisions (also referred here as partitioning decisions) impact (i) the application performance and its quality, (ii) the overall development process, and (iii) the application/system lifecycle management.

In traditional development processes, at design phase a separation into two design flows occurs: i.e. the hardware and software design flows. These latter usually evolve separately until the integration of the application during the implementation phase. In Figure 1 a simplified overview of a traditional development process is shown. In practice, the development process is more complex: phases get interleaved and several iterations and/or optimization are needed.

In this scenario, the design phase is affected by several issues which create side-effects such as hardware or/and software flow interruptions due to their mutual dependencies, redesign and unplanned iterations. These negatively impact the overall development process in terms of efficiency, quality and costs. However, most of these issues are caused by the fact that the architectural separation of the application into hardware and software occurs in an early stage of the design without the support of a systematic decision process, even though hardware and software are tightly connected in embedded applications. Usually, the design phase starts with an initial architecture’s system discussion between the hardware and software engineers, immediately after the software engineers start designing and implementing the software, and so the hardware ones, who get into the hardware design and prototyping. As a consequence, the partitioning of the application is carried out on an early stage of the design. So, from one hand the hardware design does not take into account the computational power required by software and the capability that the software might offer for enabling hardware optimization, and on the other hand the software design does not impact the hardware design and does not fully exploit the available
The integration of hardware and software is then often carried out in a late stage of the implementation phase, which does not allow performing cost effective changes.

In addition to the above, in the partitioning decisions only few factors are taken into account by the engineers, which are most related to the run-time properties (e.g. performance, resource availability, power consumption) and the decisions are most likely based upon hardware and software engineers’ expertise. However, due to the increases in complexity (i) of the applications and systems requirements as well as (ii) of the underlying platform technologies, decisions need to be accounted over a wider spectrum of properties, which spans from application/system life-cycle properties such as availability, safety, security, standard compliances, etc. to business goals and project constraints related properties such as available expertise, development lead time and costs, development of mass products or product line and so forth. Overall, this makes the partitioning process quite complex and difficult to handle/accomplish without the support of a systematic and effective methodology able to comprehensively explore many, different and even conflicting properties. Those properties are properties related to the non-functional (or extra-functional) aspects of embedded systems/applications and in this research work are referred as Extra-Functional Properties (EFPs).

1.2 Main Research Objective and Research Questions

Based on the formulation of the research problem, the **overall objective** of this research work is stated as follows:

“The design of a multiple properties-based partitioning and systematic methodology for HW and SW deployment of embedded applications into heterogeneous platforms”.

![Figure 1 - Traditional development process and related issues.](image-url)
The overall research objective is motivated by the fact that today’s partitioning approaches lack a systematic and efficient methodology able of (i) pushing the partitioning into a late stage of the design phase, (ii) defining a process to supporting engineers before carrying out the partitioning and (iii) accounting decisions based on many criteria derived by the application/system requirements as well as the business goals and project constraints.

Driven by the overall objective, the main research questions are formulated as follows:

Q1. *What would be a technology-independent process that enables the application partitioning in a late stage of the design?*

Q2. *How to enable a systematic and efficient decision process that supports the engineers in performing the application partitioning?*

Q3. *How to obtain an optimal and sustainable partitioning solution(s) which taken into account decisions related to systems requirements, business goals and project constraints?*

In Q3, by the word sustainable we mean a partitioning solution able of supporting/facilitating the application sustainability over the entire lifecycle. By the word optimal we mean a solution able to trade-off the systems requirements and satisfy the performance constraints such as execution time, lower power consumption.

The aforementioned research questions are driving a long-term research project where the final outcomes are expected to be provided for the doctoral thesis. However, with respect to the main objective, in this licentiate thesis we will provide a proof-of-concept and evaluate the feasibility of the proposed methodology.

### 1.3 Research Methodology

In order to answer to the main research question, our approach is of applied nature. Essentially, we conceived a new partitioning methodology. To evaluate and assess its feasibility we focused on building a proof-of-concept, simulating and verifying the methodology through an industrial prototype.

Specifically, depending on the different stage of project, we have applied several research methods:

A. **Analysis**

   - survey of literature and of the current state of practice
   - on existing partitioning approaches (in literature and in industry)
   - on Multiple Criteria Decision Analysis (MCDA) methods and related existing software

B. **Research Problem Identification** which leads to the

   - definition of the key research objective and questions
C. **Construction of a new approach** which includes the

- definition and design of the partitioning process flow and related activities
- development of HW/SW component-based metamodel for enabling the partitioning
- Identification of a MCDA tool chain supporting the partitioning

D. **Validation**

- **Interview:** design, realization and analysis of interview to analyze of HW/SW partitioning impact on Extra-Functional Properties (EFPs) in the industrial domain
- **Case Study:** design and semi-automatic implementation of a wind turbine application used to build an industrial demonstrator

To perform the research, our strategy follows the guidelines proposed by Shaw in [51]. The research strategy is depicted by Figure 2 and described as follows:

![Figure 2 - Research Strategy Overview](image)

- **Analysis and Identification of the Practical Problem:** we have analyzed and discussed what the practical problem is and the related issues which are associated to a typical industrial embedded design process. We reached this through the analysis of the state of the art and practice. Based on this, we narrowed down the scope and we identified the main research problem: “*today partitioning approaches are not designed to (i) perform a late and implementation-independent partitioning and (ii) to take into account EFPs which are derived from runtime and lifecycle requirements, business goals and project constraints*”. Consequently our ultimate goal is to develop such a method that will solve this problem.

- **Definition of the Idealized Problem:** We mapped the practical problem into a research setting context. In this context, we formalized the problem into a number of key research questions. In order to answer to the research questions, we carried out further literature studies and investigation analysis. They were focused on existing partitioning approaches and MCDA methods.

- **Development of Research Products:** The main outcomes of our research was the definition of a MCDA-based partitioning decision process, a set of metamodels for enabling the partitioning accounting both (more details are provided in Section1.4), the definition and implementation of an integration
framework for supporting the design phase and the partitioning process, and a set of guidelines to handle EFPs in the Automation and Control domain.

- Definition and design of the **Solution to the Idealized and Practical Problem**: We defined, designed and provided a solution, in form of a new partitioning methodology able of answering to research questions. We proposed it as solution to the idealized problem. In order to provide a step forwards to the Solution to the practical problem, we carried out an interview-survey in the Automation and Control domain.

- Research **Validation**: In order to validate the proposed idealized solution, our strategy was based on questions related to the feasibility and the efficiency of the proposed methodology: *Is it possible to do achieve our main research goal* (feasibility-question)? To answer to this question, we validate the proposed methodology by the development of a case study which was targeting the deployment of an industrial prototype.

In order improve, strengthen and complement the proposed methodology with respect to the *Solution to Practical Problem* and since the overall research will not be concluded with this licentiate thesis, we will deliver further *Research Products* as well as we will perform further *Validations* towards the development of a real product.

### 1.4 Contribution

The main contribution of the thesis is here summarized as follows:

*The definition of a new and systematic methodology enabling a late and technology-independent design partitioning of embedded application on heterogenous platforms. The proposed methodology is able to (i) find an optimal solution(s) which takes into account both application/systems requirements as well as business goals/project constraints, and (ii) facilitate/enable reuse.*

Specifically in this licentiate thesis we provide:

A. The design of the partitioning process flow, i.e. the set of activities and artifacts enabling the partitioning.

B. The specification of a metamodel suitable for both describing HW and SW components as well as their related EFPs which is used for enabling the partitioning

C. The categorization of HW and SW component EFPs in respect to the partitioning and analysis of HW/SW partitioning impact on EFPs for the Control and Automation domain.

D. The suitability assessment of MCDA methods for achieving the partitioning solution (or collection of).
E. A seamless integration of a MCDA methods and tools able of supporting the proposed partitioning process into the iFEST1.

With respect to a traditional development process, the proposed methodology allows to achieve the implementation independent-partitioning of the application in a late stage of the design as be depicted as Figure 3. Additionally, it allows to reach optimal partitioning solution taking into account multiple and even conflicting decision criteria by applying MCDA techniques. As a consequence, a partitioning solution (or collection of) is reflecting decisions which do not only account a limited number of physical/run-time EFPs (e.g. performance-related) but do also take into account many and diverse life-cycle and business goals- and project constraints -related EFPs. The proposed methodology is named MultiPar which stands for Multiple properties-based Partitioning.

![Figure 3](image.png)

**Figure 3 - Simplified overview of the proposed partitioning methodology**

Specifically, our work presents a new and systematic methodology which enables technology-independent design in an early stage of the design and reusing existing solutions by combining Component- and Model-based techniques.

More precise our contribution consist of:

1. A novel part of our research work is the formal definition of a comprehensive metamodels able of (i) describing both hardware and software components and related EFPs, (ii) supporting multiple EFPs-based partitioning and (iii) enabling reusability. As main objective we provided a theoretical model of component-based systems (CBS) which allows to specify both hardware and software components and to capture and handle their related EFPs. We

---

1 iFEST – industrial Framework for Embedded Systems Tools (http://www.artemis-ifest.eu/). It is an Artemis JU research project which goal was to specifying and developing a tool integration framework for HW/SW co-design of heterogeneous and multi-core embedded systems. I have worked on this project to develop an iFEST- compliant tools platform instance.
extend and adapted some existing software component models. The extension was necessary in order to be able to specify the hardware components as well. CBS is today a well-established technology to enable reuse of already existing components; we used such capability and strengthened it in our approach by including a library of reusable components which have to be conformed to the aforementioned component and properties metamodels.

2. An analysis of EFPs with respect to component hardware and software deployment. We analyzed several standards and existing quality models focused on EFPs, together with the support of several experts. This results into a categorization of EFPs in respect to the partitioning. Based on this categorization, we then carried out an interview-survey with researchers and industrial senior designers in the area of Automation and Control domain, which the main goal was to assess the impact of the EFPs with respect to the partitioning decisions.

3. A novel partitioning process flow and related activities which are based on the following strategy:
   (i) enabling application technology-independent design in the early stage of the design phase and pushing the partitioning decisions for enabling platform-specific design to a late stage
   (ii) enabling the application deployment into software and hardware based on multiple criteria decisions which takes into account the high level systems requirements (both runtime and lifecycle), business goals and project constraints
   (iii) making decisions based on a reusable set of alternatives, in order to enable design and implementation reusability.
   (iv) defining a seamless integration of a MCDA methods and tools into the iFEST. This part of the work includes the specification of (i) partitioning tools integration scenarios with other development tools and utilization scenarios.

Finally, we show the feasibility of the proposed platform as well as of the proposed partitioning methodology through the realization of an industrial prototype.

A more detailed description of the contributions with respect to the research questions is provided through the presentation of the peer-to-peer reviewed papers.

**Paper I (Published)**


Abstract— Many types of embedded systems applications are implemented as a combination of software and hardware. For such systems the mapping of the application units into hardware and software, i.e. the partitioning process, is a key phase of the design. Although there exist techniques for partitioning, the entire process, in particular in relation to different application requirements and project constraints, is not properly supported. This leads to several unplanned iterations, redesigns and interruptions due to uncontrolled dependencies between hardware and software parts. In order to overcome these problems,
we provide a design process that enables the partitioning based on a multiple criteria decision analysis in a late design phase. We illustrate the proposed approach and provide a proof-of concept on an industrial case study to validate the approach applicability.

This work answers to the first research question (Q1) through the specification of a new methodology able of systematically partitioning the application in a late stage of the design phase.

My Contribution - With respect to the definition of the new partitioning methodology all authors have equally contributed. However, I was responsible for the detailed definition of the partitioning decision process, the establishment of the design tool chain and the demonstration of the process viability through the realization of a case study.

Paper II (Published)


Abstract—In many embedded systems types the separation process for deploying the applications as software and hardware executable units, called partitioning is crucial. This is due to the fact that partitioning decisions impact the overall life cycle of the systems. In industry it is common practice to take partitioning decisions in an early stage of the design, based on hardware and software designers’ expertise. We propose a new methodology as a combination of model-based and component-based approaches which enables late partitioning decisions based on high level system requirements and project constrains. The final partitioning is decided based on a multi-property analysis approach. Here, we focus on the formalization of the overall process and in particular on the definition of a comprehensive system metamodel. This is meant to support modelling approaches suitable for enabling both the partitioning and reuse. An industrial case study is used to illustrate the approach.

This work contributes in answering to Q1 and Q2 by the definition of the process flow activities and the specification of a comprehensive component-based system metamodel. More specifically, with respect to the second research question (Q2) it provides means able to systematically and effectively support designers before performing the partitioning.

My Contribution - I was responsible for the detailed specification of the metamodel and the definition of process flow activities enabling the partitioning. In addition, I show the conformability of the proposed metamodel on a real industrial application.

Paper III (Submitted)

Abstract—The growing advances in hardware technologies enables significant improvements in application performance by deployment of components to dedicated executable units. This is in particular valid for Cyber-Physical Systems in which the applications are partitioned in HW and SW execution units. The growing complexity of such systems, and increasing requirements, both project and product-related, make the partitioning decision process complex. Although the different approaches of this decision process have been proposed during the last decades, they lack the possibility of providing relevant decisions based on a larger number of requirements and project/business constraints. A sound approach to this problem is taking into account all relevant requirements and constraints and their relations to the properties of the components deployed either as HW or SW units. This must be done in a systematic way, using a formal and automated technique, such as multi-criteria decision analysis. This in its turn requires formal definitions of the applications as compositions of HW and SW components, and an impact analysis of the component properties in respect to their HW/SW deployment. The aim of this paper is twofold: a) providing a formal architectural model of component-based applications with specifications of their properties with respect to their partitioning, and b) categorization of component properties in relation to HW/SW deployment. The latter is demonstrated through a property guideline to the partitioning for the System Automation and Control domain. The guideline is based on the interviews with the practitioners and researchers, the experts in this domain.

With respect to Q2 the work provides a categorization of component properties in relation to HW/SW deployment which includes lifecycle properties, runtime properties as well as business goals and project constraints-related properties.

My contribution - it is summarized as follows: the modeling of the component-based system and of the component EFPs, the categorization of partitioning-related properties, the design and realization of the interview-survey as well as the analysis of the results, in order to provide the property guideline to engineers working in the System Automation and Control domains.

Paper IV (To be submitted)


Abstract—(not available yet)

Contribution – The work will present an investigation MCDA methods and an analysis of these methods with respect to the requirements on the partitioning. It will also provide an assessment of MCDAs methods suitability and applicability in multiple EFPs-based partitioning. In addition, (i) a design of the process flow for performing the partitioning using the selected MCDAs method (or collections of) will be provided, (ii) the design of tool integration scenarios as well as the definition of a tool chain which is using the tools related to the selected MCDA methods is provided. The feasibility of the designed process flow using the established tools chain will be proven through an industrial demonstrator and it will be based on the EFPs guidelines obtained in the interview-survey (see paper III).
In the work the overall design methodology is designed, as a consequence it contributes to answer the first research question (Q1). The main contribution of this research work is provided with respect to the Q3 with respect to the MCDA techniques and on how to use this techniques for enabling the partitioning by taking into account many EFPs.

*My Contribution* — I managed and guided the investigation of the MCDA methods and related tools. I defined the partitioning requirements on MCDA methods and tools to perform the analysis and assessment of their suitability for the MultiPar methodology. I supervised and guided the definition of the process flow. I designed the tools chain and the integration scenarios, and worked on the realization of the demonstrator (modeling, design and implementation).

**Paper V (Published)**


**Abstract**— Tool integration in the context of embedded systems development and maintenance is challenging due to such systems' lengthy life cycles and adaptability to process specifications. The iFEST project provides flexibility in development processes and extends support for long product life cycles.

This work focused on providing an answer to the second research question (Q2). Specifically, it defines a tool chain which is seamlessly integrated and orchestrated over an integration platform. This latter allows to guarantee efficiency, systematicity to the engineers during the development in general and at the design phase. It has been demonstrate through the design of an industrial application.

*My Contribution* — I was responsible for (i) defining the orchestrator specifications, (ii) the definition of the design as well as (iii) the partitioning integration scenarios over the platform, and (iv) the design and realization of the industrial application to be deployed on a heterogeneous (hardware) platform.
2 Thesis Outline

The licentiate thesis will be presented as a collection of most relevant published papers. In order to present the research work the first part of the thesis will provide an introduction, research overview (including the problem description, the main research objective and methodology) and summarizing discussion and conclusion. The second part will include the papers that have been peer reviewed and published in the academic community. They are referred to in the text using Roman numerals.

The following outlines the sections of the licentiate thesis:

   Abstract
   List of Included Research Work
   List of Not Included Research Work

Part - Thesis

1. **Introduction**: it presents the general description of the context in which the research is carried out, specifically it will focus on the role of the partitioning in embedded systems design (in the past and today) and it will describe the background and motivation to the research project.

2. **Research Summary**: it presents the research problem, the objective, the methodology undertaken and how the achieved results are validated. It is divided in sub-sections as follows:

   2.1 **Problem Description**: it describes the current state of practice and the research problems related to the partitioning (see Section 1.1 of this proposal)

   2.2 **Research Objective and Research Questions**: it will presents the main research objective and the main research questions, as outlined in Section 1.4 of this proposal

   2.3 **Research Methodology**: it will provide a discussion on the undertaken research approach, and a more in detail it will provide the description of the specific methods used for the different stages of the research, as presented in Section 1.4 of this proposal.

   2.4 **Research Validation**: it will discuss how the methodology has been validated, it will focus on the proof-of-concept related to the industrial case study.

3. **Contribution**: It summarizes the findings and conclusions from the included papers as outlined in Section 1.4 of this proposal.

4. **Related Work**: it presents relevant fields of research and practice. The related work will be divided as follows: Hardware/Software Partitioning; Component Modelling and EFPs for Embedded Systems, and Multiple Criteria Decision Analysis.

5. **Conclusion**: it summarizes the licentiate thesis and discusses the future research directions.
6. **References**: it lists the references used in the thesis.

**Part - Included Papers**

- **Paper I** - *Partitioning Decision Process for Embedded Hardware and Software Deployment*
- **Paper II** - *Modelling for Hardware and Software Partitioning based on Multiple Properties*
- **Paper III** - *Architectural Decisions for HW/SW Partitioning based on multiple Extra-Functional Properties*
- **Paper IV** - *Assessing Multiple Criteria Decision Analysis Suitability for HW/SW Architectural Partitioning in Embedded Systems Design*
- **Paper V** - *A Tool Integration Framework for Sustainable Embedded Systems Development*
3 Related Work

The related work is divided as follows:

**Hardware/Software Partitioning.**

In literature, hardware/software partitioning (also referred as partitioning for brevity) is a widely discussed research topic in the context of the co-design, e.g. [1], [2], [3], [4], [5]. According to two of the most a relevant contributions in this field, the partitioning problem is “stated as finding those part of the model best implemented in hardware and those best implemented in software” [1] and “choosing the software and hardware implementation for each component of the system specification” [5]. Another definition is given in [6] which defines the hardware/software partitioning as “the problem of dividing an application’s computations into a part that executes as sequential instructions on a microprocessor (the “software”) and a part that runs as parallel circuits on some IC fabric like an ASIC or FPGA (the “hardware”), such as to achieve design goals set for metrics like performance, power, size, and cost.”

Hardware/software partitioning is considered to be one of the main challenges when designing embedded applications: “as it has a crucial impact both on the cost and the overall performance of the resulted product” [7]. Hardware/software partitioning is a NP (non-deterministic polynomial)-hard problem [8]. A rigorous mathematic formulation and analysis of the partitioning problem is provided in [9], and a formal definition in form of task graphs, widely used in partitioning representation is presented in [8].

There is a considerably body of knowledge related to the partitioning problem. It has been intensively studied in the 1990s and the early years of the 2000s, when basically the co-design emerged as a new discipline [2]. Over the decades, a wide range of approaches have been proposed in order to automate/support the hardware/software partitioning using different strategies, for instance dynamic programming [10], heuristic algorithms based on the tabu search, simulated annealing, genetic algorithm techniques [11], [12] [13], [14] etc., integer programming [4], multiple objective optimization techniques [15], [16] and etc. An in-depth study of several partitioning approaches and related issues is provided in [14] and a walk through the highlights of partitioning approaches over the last two decades be found in and [2]. However, all of these approaches are mainly oriented towards solutions satisfying physical performance requirements. They provide partitioning solutions which results are derived by decisions considering platform-related indicators, like potential speedups, area, communication overheads, locality and regularity of computations [17]. Even though, there exists few partitioning approaches which optimize combination of extra-functional requirements (e.g. such design costs, energy consumptions, performance, etc.) [18], [19], they are still limited in the total number of EFPs. Additionally, current partitioning approach focused only on technical issues and do not account properties-related to the project development and business perspective, neither the EPFs of interested for the development of embedded industrial application such as safety, reliability, security, sustainability and so forth.

In order to provide an answer to (i) the increase in complexity of the embedded systems and (ii) the advances in the underlying hardware technologies when it comes to partitioning decisions, in difference to existing approaches we propose a general and systematic methodology able of partitioning
When performing MCDA-based partitioning of component-based applications, it is of key importance how components and related EFPs are modelled. Hardware and software components have to be technology-independent modelled, and the handling of different EFPs associated to a component variant (i.e. a specific component implementation, components may have associated several variants) has to be properly managed in order to enable reuse.

There exist several component models that are specified by means of metamodels, for example Palladio [20], Pecos [21], Save CCM [22] and Pro-2Com [23] which in a similar way defines interfaces and EFPs. Our approach extends these specifications by (i) allowing the description of component independently from the underlying platform and (ii) allowing variable management of EFPs with respect to different component variants. Some component models allow different implementations of components, but these implementations use the same technology and have the same EFP types for all variants, though with different values. By allowing a variable set of EFPs for component variants we have made it possible to reason about very different implementations, like software and hardware.

The “Management of extra-functional properties in component models is one of the main challenges in the component-based software engineering community” [23]. In the last years, several research efforts have been spent in order to support the management of extra-functional requirements in embedded systems development (e.g., [24], [25], and [26]), from both hardware and software perspectives. In particular, different techniques have been introduced in order to support the specific features of an embedded system (e.g., the analysis of timing properties that are typically computational and time consuming [27], or the management, the preservation and the analysis reuse of EFPs that are also critical tasks [28], [29]). Component models have been introduced for supporting the embedded systems’ development (e.g., [22], [30], and [31]), and approaches for embedded systems’ adaptation under EFPs constraints have been proposed (see, e.g., [32], [33]). To estimate EFPs for software and hardware implementations approaches have also been introduced (see, for example, [34], [35], [36], for power and energy estimation). The run time of a hardware implementation on a FPGA, for example in [37] is estimated by exploiting the simulation and a performance model of the FPGA. On the contrary, a statistical approach is proposed in [38] in order to estimate the execution time of embedded software.

The topic of the definition of metamodels to specify EFPs has been also studied intensely. For example, in [39] fault tolerance aspects have been covered by using a metamodel, while in [40] a service-oriented metamodel for distributed embedded real-time systems covers real-time properties of services, like response time, duration, deadline. In the latest decades, digital and software designs methodologies have become more alike [1], and as already foresaw in [41] they require designers to have a unified view of software and hardware, which converges to the concurrent design of hardware/software [1] (see, for example, the HWSWCO3 project [50] and the co-design framework in [42]). We have extended the current research state of the art by providing a set of metamodels able of (i) describing embedded component-
based applications from both hardware and software perspectives, as well as (ii) capturing systems (from lifecycle and runtime perspectives) and business goals and project constraints related EFPs.

**Multiple Criteria Decision Analysis.**

Multiple Criteria Decision Analysis (MCDA) is a sub-discipline of operational research, which objective is “support the subjective evaluation of a finite number of decision alternatives under a finite number of performance criteria, by a single decision maker or by a group.” [43]. In [44], MCDA is defined as “an umbrella term to describe a collection of formal approaches which seek to take explicit account of multiple criteria in helping individuals or groups explore decisions that matter”. In [45], Figueira at al. contribute to the state of the art with one of the most well-known survey on MCDA. Since the 60s, MCDA is widely applied in several fields such as medicine, human resources management, pattern recognition, marketing, financial management and economics, environmental and energy management, computer science, etc. [46], [45] to solve complex decision problems.

A formal definition of a typical MCDA problem is provided in [47]. There exist several MCDA methods which presented in [43] MCDA methods” have been designed in order to designate a preferred alternative, to classify the alternatives in a small number of categories, and/or to rank the alternatives in a subjective order of preference”. The key aspects of MCDA and related methods it is that it allows to “take the multidimensionality of decision problems into account by using multiple criteria, instead of one common denominator, such as a monetary number in Cost-Benefit Analysis” [48]. It can be used for dealing with many, heterogeneous and even conflicting criteria as a consequence we consider MCDA suitable for enabling partitioning based on several and of different nature criteria (i.e. derived by the EFPs related to the application requirements and business goals and project constraints-related constraints). Even though it is limited to just few as well as performance-related EFPs, an example of research work on this direction is given by [49], where MCDA methods are used to ranking different hardware and software design choices.
4 Current Status and Time Plan

4.1 Publications
Below, a list of the publications is presented. It is divided into the following categories (i) papers whose contribution is fundamental for the thesis and (ii) papers and technical reports that are related to the thesis.

4.1.1 Publications Fundamental to the Thesis Contributions

Conferences and Workshops


Journals & Magazines


4.1.2 Publications Related to the Thesis

Conferences and Workshops


MRTC Technical Reports
4.2 Courses

In Table 1 the list of the activities and courses completed and approved it is provided. The number of credits required for the Licentiate Thesis is reached (45 ECTS).

<table>
<thead>
<tr>
<th>ID</th>
<th>Course</th>
<th>Credits (ECTS)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Master thesis &quot;language and tool for modelling and implementation of motion control in heterogeneous systems&quot;</td>
<td>15</td>
<td>Completed</td>
</tr>
<tr>
<td>2</td>
<td>Process Control</td>
<td>2.5</td>
<td>Completed</td>
</tr>
<tr>
<td>3</td>
<td>Computers in Industry</td>
<td>4.5</td>
<td>Completed</td>
</tr>
<tr>
<td>4</td>
<td>Computer Architecture II</td>
<td>4.5</td>
<td>Completed</td>
</tr>
<tr>
<td>5</td>
<td>Modelling in UML &amp; Simulink</td>
<td>2</td>
<td>Completed</td>
</tr>
<tr>
<td>6</td>
<td>Research Planning</td>
<td>4.5</td>
<td>Completed</td>
</tr>
<tr>
<td>7</td>
<td>Research methods in natural science and engineering</td>
<td>7.5</td>
<td>Completed</td>
</tr>
<tr>
<td>8</td>
<td>Introduction to graduate education for new PhD students</td>
<td>4.5</td>
<td>Completed</td>
</tr>
</tbody>
</table>

**Table 1 - Completed Course List**

| Total Credits (ECTS) | 45 |

4.3 Scheduled Activities and Time Plan

The goal is to have the licentiate defense in at the end of the first quarter of 2014. In order to achieve the remaining activities to be accomplished and the time plan are presented:

**Scheduled Activities**

To complete the thesis, the main remaining activities are planned:

- Complete and submit the second journal paper (2013-12-30)
- Write the licentiate thesis.
Time Plan

A preliminary time plan is scheduled as follows:

- Licentiate Thesis proposal presentation 201-12-20
- Complete the first draft of the licentiate thesis 2014-02-10
- Complete the licentiate thesis and send it to opponents 2014-02-28
- Finish camera-ready copy of the licentiate thesis 2014-02-28
- Licentiate Defense 2014-03-31

Potential opponent and grading committee members are:
- Mehmet Aksit - Professor at Computer Science University of Twente - Netherlands
- Ingo Sander - Associate Professor at the Royal Institute of Technology KTH – Sweden
- Vasilios Chouliaras - a senior lecture at the Loughborough University - England
- Lars Asplund, Professor at the Mälardalen University (MDH) - Sweden
- Martin Törgren - Professor at the Royal Institute of Technology KTH – Sweden
- Michel Chaudron – Professor at Chalmers University of Gothenburg - Sweden
- Per Stenström Chaudron – Professor at Chalmers University of Gothenburg - Sweden
- Raffaela Mirandola Assistant Professor at the Politecnico di Milano - Italy
- Petr Tuma – Associate Professor at Charles University – Czech Republic
- Sahra Sedigh – Associate Professor at Missouri University of Science and Technology – US
- Elena Dubrova - Professor at the Royal Institute of Technology KTH – Sweden
- Detlef Streitferdt – Junior Professor at University of Waterloo - Germany
5 References


