Multi-objective Design Space Exploration based on UML

Marcio F. da S. Oliveira, Eduardo W. Brião, Francisco A. Nascimento, Lisane Brisolara, Luigi Carro, Flávio R. Wagner
Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil
{mfsoliveira, ewbriao, fanascimento, lisane, carro, flavio}@inf.ufrgs.br

Abstract. This paper proposes a Design Space Exploration (DSE) framework using UML-based estimation and a multi-objective design exploration mechanism. This framework allows the designer to automatically select the most adequate modeling solution for application, architecture, and mapping, in an integrated and simultaneous way and at a very early design stage. An MDA-based transformation engine implements the mapping by model transformations, allowing the evaluation of possible implementations during a DSE process based on accurate UML-based estimates of the effects of each transformation, without requiring costly hardware synthesis and code generation steps. A design space exploration scenario for the design of a real application has been developed. Experimental results show that the proposed estimation and exploration approach may find a suitable solution regarding the design requirements and constraints in a very short time, with an acceptable accuracy, without relying on costly synthesis-and-simulation cycles.

1 Introduction

Embedded systems design has become mainly software-driven. For each new embedded product more functionality is demanded and more complex software components are added to the system. With the rising complexity of software, and benefiting from advances in technology, MPSoC (Multi-Processor System-on-Chip) platforms have emerged as the only feasible solution to satisfy the strong constrains imposed to embedded system design. To deal with the ever-growing challenge in this area, two orthogonal approaches have been proposed. The first approach is Platform-based Design (PBD) [2], which is a successful meet-in-the-middle strategy that is used to maximize the reuse of pre-designed components and achieve the best customization of the design according to the system requirements. In this approach, an architecture is an instance of a generic platform, tailored for application-specific requirements. The second approach is the Model Driven Development, in which UML and Model Driven Architecture (MDA) are used to raise the abstraction levels and to provide mechanisms to improve the portability, interoperability, maintainability, and reusability of models. A fundamental concept closely related to the PBD approach is the orthogonalization of concerns, where system function and architecture are independently developed, and a later mapping between them tries to match the design
requirements. This also allows easy retargeting of an application into different architectures.

Moving the development focus from implementation to model suggests the support to a fast design space exploration (DSE) in the early design steps, where the design effort is now concentrated and modeling decisions can lead to substantially superior improvements. In this context, this work proposes a DSE framework using UML-based estimation and a multi-objective design exploration mechanism, which allows the designer to automatically select the most adequate modeling solution for application, architecture, and mapping, in an integrated and simultaneous way and at a very early design stage.

Experimental results, extracted from a design space exploration scenario for the design of a real application, show that the proposed estimation and exploration approach may find a suitable solution regarding the design requirements and constraints in a very short time, with an acceptable accuracy, without relying on costly synthesis-and-simulation cycles. Besides, the use of UML and MDA promotes the reusability of application and architecture models.

The remaining of this paper is organized as follows. Section 2 compares our methodology to other model-based approaches. Section 3 presents the meta-modeling infrastructure and transformation engine, which allows the model-based DSE, and explains how this infrastructure is used by our design space exploration method. A real case study, which illustrates and validates our DSE approach, is described in Section 4. Finally, Section 5 draws main conclusions and proposes future research directions.

2 Related Work

There are many recent research efforts on embedded systems design based on MDA. The adoption of platform-independent design and executable UML has been vastly investigated. For example, xtUML [4] defines an executable and translatable UML subset for embedded real time systems, allowing the simulation of UML models and the code generation for C oriented to different microcontroller platforms. The Model Execution Platform (MEP) [5] is another approach based on MDA, oriented to code generation and model execution. Our approach is more related to the design space exploration at a high abstraction level, using the notion of transformations between models, than to the validation or code generation, which are the main concerns of these approaches.

The Generic Modeling Environment [6] (GME) is a modeling tool oriented to the development of domain-specific visual modeling languages by the specification of a meta-model. In GME, meta-models are described as UML class diagrams, and, once created the meta-model for a specific domain, a visual editor is automatically generated that supports the edition of diagrams for the domain as well as the generation of design artifacts. GME has been used to implement some model-based tools for embedded system design. MILAN [1] (Model-based Integrated Simulation) is one such tool, where the main focus is on the simulation of embedded systems.
Multi-objective Design Space Exploration based on UML

The DaRT [7] (Data Parallelism to Real Time) project also proposes an MDA-based approach for SoC (System-On-a-Chip) design that has many similarities with our approach in terms of the use of meta-modeling concepts. The DaRT work defines MOF-based meta-models to specify application, architecture, and software/hardware association and uses transformations between models as code transformations to optimize an association model. In doing so, it allows the re-factoring of an application model in order to better match it with a given architecture model. In DaRT, no design space exploration strategy based on these transformations is implemented, and the main focus is mainly the code generation for simulation at TLM (Transaction Level Model) and RT (Register Transfer) levels.

Compared to our approach, no related work takes advantage of the MDA notion of transformation between models to represent, delimitate, and explore the design space of possible implementations for a given application on a specified platform at a high abstraction level.

3 Model-based Design Space Exploration for Embedded Systems

In our approach, the UML models of the application, platform, architecture, and mapping follow the MDA approach, in which meta-models for each dimension are defined according to the MOF (Meta Object Facility) standard and handled through the Metadata Repository (MDR) API. A meta-model is also available for the definition of design constraints and requirements to be met by the application-architecture mapping.

At the beginning of the development process, the designer may investigate the distribution of responsibility between classes, the encapsulation of functionality inside objects, the aggregation of objects inside independent threads, and strategies for the interaction between objects. A UML model of the application captures these modeling decisions. An architectural platform is also defined in UML, mainly using class diagrams and according to the platform meta-model. Also using UML modeling resources, the possible mappings between the elements of the application and of the platform are specified. These mappings suggest the possible partitioning of functions into tasks and the binding of these tasks into resources of the platform.

Our meta-data repository contains information on meta-model definitions and also on specific models for hardware and software components and resources, to be reused in the definition of applications and platforms, as well as information on the various models created during the design process. The reused components and resources are previously characterized regarding their costs in performance, memory footprint, and energy consumption. This characterization is performed with appropriate off-line tools, and the information is incorporated in the meta-data repository. In the case of software components, their costs when mapped into available hardware components are available in the meta-data repository.

Our tool for the DSE process adopts a heuristic approach to investigate the space of possible implementations, which is based on simulated annealing. The exploration must find a solution with minimal communication cost and minimal energy consumption, under hard real-time constraints, meeting design restrictions such as the
Marcio F. da S. Oliveira, Eduardo W. Brião, Francisco A. Nascimento, Lisane Brisolara, Luigi Carro, Flávio R. Wagner

processing capacity of each processor (which depends on its frequency/voltage setting).

In a first prototype implementation, our exploration tool selects the most appropriate solution regarding three design decisions: the partitioning of tasks between processors (given a fixed number of processors), the mapping of processors into a segmented bus, and the voltage scheduling for each processor. At each moment, the tool takes some design decisions and incorporates the corresponding information in the conditions of the transformations rules of the mapping model.

At this point, our transformation engine is invoked to traverse the set of transformation rules, so as to select and apply the ones that are enabled at that moment in order to produce a hardware/software mapped architecture in the form of an implementation model. This model is the input to the SPEU estimation tool (System Properties Estimation with UML) [3]. SPEU supports model evaluation for early and fast design space exploration, providing analytical estimates about physical system properties (execution cycles, energy consumption, volume of communication data, and memory footprint), which are directly obtained from instances of the proposed infrastructure meta-models, with reasonable accuracy when the reuse of repository components is largely employed by the PBD. The estimation is based on the platform model information and on an ILP (Integer Linear Programming) formulation for the identification of best-case and worst-case execution paths. This allows the exploration tool to evaluate each candidate solution during the DSE process, without depending on costly synthesis-and-simulation evaluation cycles.

4 Case Study

A design space exploration scenario for the design of a real application, concerning the automated control of a wheelchair with functions such as movement control, collision avoidance (ultrasound and stereo vision), and navigation, has been developed. The application model contains 17 tasks, which have communication dependence with other ones. The platform model is based on the femtoJava microcontroller [9], upon which an API and operating system components [8] for real-time behaviour and communication support is available. An architecture containing 4 processor units, which are distributed between two bus segments, was previously selected. Fig. 1.a illustrates the task graph extracted from the Application Model, and Fig. 1.b illustrates the previously selected architecture.

The DSE tool partitions tasks between processors in order to minimize the overall communication costs between tasks. Tasks with a higher communication bandwidth between them tend thus to be grouped in the same processor, if possible. The candidate task partitioning can neither violate the task deadlines nor exceed the processing capacity of each node. Simultaneously, the exploration tool maps processors into the two bus segments, such that the overall communication cost is minimized. Therefore, processors with a higher communication between them (because of the tasks assigned to each one) tend to be mapped into the same segment. Moreover, in order to further reduce the energy consumption, the exploration tool also selects the minimal frequency for each processor, but avoiding task deadline
Multi-objective Design Space Exploration based on UML

violations. Realistic energy, performance, and memory costs for HW and SW platform components and services (such as task scheduling) have been used to guide the exploration using the SPEU tool. Table 1 presents the 5 best candidate solutions found during the DSE process.

![Application Model task graph](image1)

![Previously selected architecture](image2)

Fig. 1. a) Application Model task graph; b) the previously selected architecture

Table 1. The 5 best candidate solutions obtained by the DSE exploration tool

<table>
<thead>
<tr>
<th>Solution</th>
<th>Energy (mJ)</th>
<th>Communication (Bytes)</th>
<th>Performance (cycles)</th>
<th>Memory (Kbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>484,918.688</td>
<td>614,403.000</td>
<td>21,418.438</td>
<td>385,671.849</td>
</tr>
<tr>
<td>2</td>
<td>484,878.875</td>
<td>614,403.000</td>
<td>21,418.131</td>
<td>385,671.849</td>
</tr>
<tr>
<td>3</td>
<td>483,752.603</td>
<td>307,207.000</td>
<td>21,417.824</td>
<td>385,671.542</td>
</tr>
<tr>
<td>4</td>
<td>482,884.546</td>
<td>307,206.000</td>
<td>21,375.729</td>
<td>385,671.542</td>
</tr>
<tr>
<td>5</td>
<td>482,883.510</td>
<td>307,206.000</td>
<td>21,375.729</td>
<td>385,671.542</td>
</tr>
</tbody>
</table>

The DSE tool tried to optimize energy, communication amount, performance, and memory. The priority was to reduce the energy consumption. As the DSE tool must respect the task deadlines, performance had a low weight in the design optimization. Table 1 shows that the DSE tool could not reduce the communication amount in the buses to less than 307,206 bytes. The design restrictions, such as deadlines and processor capacities, were respected by the DSE tool, so that it could not map more tasks into the same processor. For improving the energy savings, when task mapping and processor allocation could not further reduce the energy, the DSE tool reduces the processor frequency, also avoiding task deadline violations. Thus, for the same mapping and allocation (compare solutions 4 and 5), the improvement (for solution 5) is only in the energy and comes from a different processor voltage setting. The communication amounts presented by solutions 1 and 2 are two times larger than other ones. The reduction in the communication amount is due to a better task mapping, which grouped more communication tasks into the same processor.

This case study illustrates a design space exploration scenario where a real application presenting a large design space was designed. Through the Multi-objective Design Space Exploration approach present in this work, the designer can specify the system using UML models at high abstraction level and look for the best alternative to be implemented without relying on costly synthesis-and-simulation evaluation cycles.
5 Conclusions and Future Work

A design space exploration approach based on UML/MDA was presented, which adopts a meta-modeling infrastructure with specific meta-models for application, platform, mapping, and implementation models. The MDA fundamental notion of transformation between models is used to represent and delimitate the design space of possible implementations of the specified application on the specified platform. A set of transformation rules defines the possible mappings from application into platform, and by evaluating these rules the SPEU tool obtain estimates for the possible implementations without any costly hardware and software synthesis/simulation steps.

Experimental results show that the proposed estimation and exploration approach may find a suitable solution regarding the design requirements and constraints in a very short time, with an acceptable accuracy, without relying on costly synthesis-and-simulation cycles. Besides, the use of UML and MDA promotes the reusability of application and architecture models.

One of the future directions to be considered is the meta-modeling improvement, in order to represent more complex applications, platforms, and mappings. Moreover, we intend to test different optimization algorithms in order to improve the design space search.

References