

Decomposition of Design Space Exploration Problems in the Context of model-based Development

Sergey Zverlov

fortiss GmbH
Guerickestr. 25, 80805 Munich, Germany
zverlov@fortiss.org
<http://www.fortiss.org/>

Abstract. During the development process of complex embedded systems there typically is a large number of architectural decisions that engineers are facing. Each of them can influence the properties of the end result (i.e.: cost, performance, etc.). A systematic process that supports this decision making is often referred to as Design Space Exploration (DSE). The need for computer-aided DSE methods was recognized by the research community which resulted in a large number of publications in this area.

The contribution of this PhD project is two-fold: First contribution is a classification of typical DSE problems that arise during the development process of embedded systems. Due their high popularity in this area the focus of this work lies on the model-based approaches. The second contribution is two provide a decomposition approach for complex DSE problems. The decomposition strategy is thereby based on the previously identified DSE problem classes.

Keywords: Design Space Exploration, Model-Based Engineering, Decomposition, Architecture Optimization

1 Introduction and Problem Statement

During the design phase of complex embedded systems engineers are faced with a number of design decisions (i.e. different selection or configuration of system components) that may influence the quality of the system under development. Depending on the size of the system the number of decisions can become very large. The traditional (manual-driven) approaches such as relying on the experience of leading engineers or following the, so-called, best practices do not always produce good results. A systematic and computer-aided approach would be much more preferable. A decision process of searching for design solutions that satisfy certain constraints (i.e.: cost, weight, etc.) within a design space is called Design Space Exploration (DSE).

The need for automated Design Space Exploration techniques was recognized by industry and academia, which led to a rising number of DSE related

publication in the last decade. The large amount of new publications and their fragmentation over different research communities led to many ad-hoc solutions (cf. [2]) that often tackle very similar problems. But given the specific nature of solutions they are very difficult to compare with each other or to reuse in a different context. Following that line of argumentation the first Research Question (**RQ1**) of my PhD project is dealing with the classification of different DSE problems. Thereby this work will focus on DSE problems that arise in the context of model-based design of embedded Systems.

Another trend, driven by the increasing computational power of modern embedded hardware, is the rising complexity of those systems. That also effects the complexity of Design Space Exploration problems, that are used during the design of these systems. Furthermore, during a preliminary study of the related literature we found that many DSE problems consist of a number of atomic sub-problems. An example for such a complex DSE problem is the search for a mapping of software to hardware together with a corresponding schedule (cf. [19]). This problem can be even extended by adding the search for an optimal selection of the hardware architecture (cf. [14]). Despite of the rapid progress in computation power and solver technologies the problems can become unsolvable at a reasonable time given an input complex enough. Therefore, the second Research Question (**RQ2**), considered in this PhD work, deals with the complexity reduction of DSE problems. Thereby the findings from RQ1 will be applied to find a decomposition strategy for complex DSE Problems that arise in the context of model-based system design of embedded systems.

2 Related Work

This work is related to two research areas: The first one deals with the generalization of DSE approaches and their classification. The generalization of DSE is often tackled by Frameworks, such as: ArcheOpterix [1], DESSERT [12], FORMULA [10], Viatra-DSE [8], CoBaSA [11] and many more. In comparison the intention of this PhD work is not to create a framework for Design Space Exploration but to define, what a DSE is and which DSE problems exist in the context of model-based development of embedded systems. The idea is that this classification can – for instance – assist tool vendors in integrating DSE methods in their products. But of course there are also recent works that target the classification of publications in this field. One of them is [2], where a taxonomy of the current works in the area of system optimization is provided. In comparison our work focuses on an abstract classification of DSE problems itself instead on the classification of publications in this area. [17] provides a classification of DSE problems. Compared to this work we provide a more abstract and concise collection of typical DSE problems.

The second related are to this work is the decomposition of complex problems into simpler sub-problems. This approach is very popular in computer science and is widely used in the field of Constraint Programming, such as Parallel Constraint Solving (cf. [6]), Distributed Constraint Solving (cf. [20]) or Geometric Constraint Solving (cf. [9]). But also in the area of Design Space Exploration

works exist which use different decomposition techniques to tackle the scalability issues. There are contributions that solve this problem by combination of different solving techniques [15], usage of tight bounds [5], iterative problem decomposition [16]. All those works use decomposition for the sake of performance increase. But their decomposition strategy is very tightly coupled with the problem under consideration. This work proposes a decomposition based on the typical DSE problems that arise during the design phase of embedded systems. We think that this way it can be applied on a larger set of problems.

3 Proposed Solution

Today's embedded systems, such as modern vehicles, are characterized by high complexity and heterogeneity. Therefore, the development process has to provide support to the engineers in dealing with these challenges. A popular approach, that is recommended by many standards and methodologies (cf. AUTOSAR [3], EAST-ADL [4], SPES Methodology [13], etc.), is the separation of concerns. Thereby the design process is decomposed into sub-processes according to physical phenomena [18], stakeholders (cf. IEEE 1471), viewpoints, levels of abstractions [13] or other criteria. But since the ultimate goal is to build a whole system those sub-processes have to be linked together (i.e.: mapping of the elements of the software viewpoint to the elements of the hardware viewpoint). Tools that are designed to support the development of embedded systems (i.e.: PREEvision) are implementing this concept.

In the following sub-sections, we discuss how it is possible to use this separation of the design process, that is dictated by many standards, to (i) define Design Space Exploration in this context; (ii) provide a classification of typical Design Space Exploration Problems; and (iii) use this classification for the decomposition of complex Design Space Exploration tasks.

3.1 Design Space Exploration in MBD

Model-based development (MBD) became more and more popular in the area of embedded systems. It is considered to be a central approach to deal with the rising complexity in this field [7]. For our classification and decomposition framework it is necessary to define Design Space Exploration for embedded Systems in the context of MBD. Even then DSE is a very broad term. So we first introduce a set of assumptions that limit the focus of this work:

Assumptions: In the context of model-based development a system consists of model elements. And each model element can have certain properties (i.e.: cost, weight, etc). So let A be the set containing all model elements of the System under Development.

As already discussed in this section, separation of concerns is a popular approach in the field of embedded systems. Therefore, we assume that A can be partitioned into disjoint sub-sets as follows:

Assumption 1 A collection of nonempty sets A_1, \dots, A_n are partitions of A iff

1. $A = A_1 \cup A_2 \cup \dots \cup A_n$
2. $\forall i, j (A_i \neq A_j) \Rightarrow A_i \cap A_j = \emptyset$

To be able to construct a valid system instance it is though necessary to provide a possibility of assemble those partitions back together.

Assumption 2 If $A_i \in A$ and $A_j \in A$ are two partitions according to ass. 1, then it is possible to find a relation $R(A_i, A_j)$ between those two partitions.

And finally it should be possible to express constraints (i.e.: derived from requirements) towards this system that consists of partitions and their relations.

Assumption 3 If A_i and R is a relation between two partitions, then is possible to express constraints of (sub-)partitions $C(A_i)$ and their relations $C(R)$.

DSE Definition Using those assumptions, it is now possible to provide a definition of Design Space Exploration in this particular context.

Definition 1. DSE is a process of systematically finding solutions from a design spaces that satisfy certain constraints. A solution thereby is either a partition, a relation between partitions or a combination of those two.

3.2 Classification of DSE Problems

Based on the assumptions and definitions from sec. 3.1 a set of abstract DSE problems is derived. We differentiate the DSE problems over their constraints. The main difference thereby is whether the constraint includes elements from one single partition or includes elements from two different partitions.

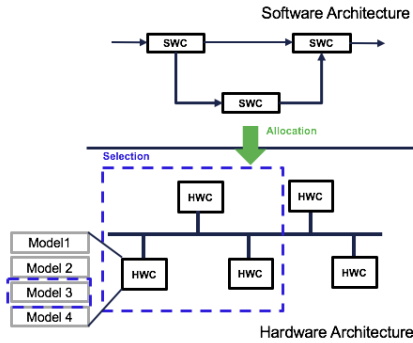


Fig. 1. Design Space Exploration Problem Classes

To illustrate this concept a small example (cf. fig. 1) is used. It shows a simplified embedded system that consists of a software and a hardware part, where the software is supposed to run on the hardware.

Definition 2. *We propose that one class of DSE problems is to select a sub-set from a partition A_i where this sub-set $A' \subseteq A_i$ conforms to a set of constraints C . This problem we call **selection**.*

An example for this class is to select features from a feature tree or components from a 150% model (cf. eq. 1). Another example is to select a configuration (or an instance) of a component (i.e. select a camera sensor model from a set).

Definition 3. *As the second class of DSE problems we propose is the search for a set of valid relations R between two partitions A_i and A_j so that $\forall r \in R(A_i, A_j) \rightarrow C(r)$.*

Examples for this operation would be the allocation (cf. eq. 2) of software to hardware (deployment) or the allocation of software onto time slots (scheduling).

We performed a preliminary study of the related literature. Based on the results we can say that most of the DSE operations within these works could be led back to the two operations from def. 2 and 3. The literature search strategy was directed towards finding published papers in journals and conference proceedings that are accessible the widely accepted search engines and databases such as Google Scholar, IEEE Xplore, ACM Digital Library etc. The search words were focused on the scope of the PhD topic, which is Design Space Exploration, model-based Development and embedded Systems.

3.3 Decomposition of DSE Problems

The idea of the decomposition approach is to divide a DSE problem into sub-problems, solve them one after another. Pruning parts of the design-space in each step and thereby reducing the overall complexity of the problem. Often the solution of one problem has an effect of the solution (or even solvability) of the other problem. If one does not take care of those dependencies it is very probable to prune to much design space during the solution of the first sub-problems. Consider the DSE problem ($P2$) from sec. 1: If during the selection of the hardware architecture to much emphasis is put on the cost-efficiency the result may not provide enough resources for a possible deployment of software to this platform in the next step. Even though the generalization of these dependencies is still subject of ongoing work, there are some first results that can be sketched here using examples. Let consider the DSE problem $P2$ again, which consists of selection ($P2.1$) and allocation ($P2.2$) according to definitions from sec. 3.2.

Let us assume that the goal of $P2.1$ is to find a hardware configuration so that its cost doesn't exceed a certain value (cf. eq. 1). For $P2.2$ we assume that each software component consumes a certain amount of memory and each hardware component provides a certain amount of memory. The goal of $P2.2$ is then to create a mapping between SWCs and HWCs. Thereby it is not allowed to exceed the memory provided by the HWCs(cf. eq. 2).

$$\forall hwc \in HWC : hwc.cost \leq value \quad (1)$$

$$\forall hwc \in HWC \sum_{swc \rightarrow hwc}^{SWC} swc.memory \leq hwc.memory \quad (2)$$

If during the solution of *P2.1* we only consider the satisfaction of eq. 1, there is a chance that in the next step (*P2.2*) we will not find a valid solution, because the cost-efficient hardware architecture does not provide enough memory. A possible way to prevent this is to add a simplified version of eq. 2 to the DSE problem *P2.1* as follows.

$$\sum^{SWC} swc.memory \leq \sum^{HWC} hwc.memory \quad (3)$$

This extra constraint ensures that the sum of provided memory is at least as high as the sum of the consumed memory. Vice versa one may conclude that if eq. 3 is not satisfied the problem *P2.2* will not have any solutions.

4 Expected Contribution

This research will provide the following scientific contributions:

1. DSE Classification Framework
 - (a) Set of DSE problem classes that are typical for the design phase of embedded systems
 - (b) Set of requirements (assumptions) a tool or methodology has to fulfill in order to be able to integrate those DSE classes
2. General Decomposition Approach for DSE Problems within Design Phase of embedded Systems
 - (a) Guide of how to decompose complex DSE Problems
 - (b) Tool Support for (semi-)automatic Decomposition of DSE Problems

5 Current State and Plan for Evaluation

At present time, we already achieved a number of tasks. We performed a literature study over the Design Space Exploration literature of the last 10 years. From that study we derived a preliminary classification of DSE problems. This allows us a decomposition of complex problems along the steps of a common development process of embedded systems. Furthermore, we conducted first experiments that showed us an increase in performance compared to traditional approaches. Using the obtained classification, we derived a first version of a Domain Specific Language that can describe the identified types DSE problems and supports our decomposition approach.

We intend to evaluate the applicability of our approach using an industrial case study. Furthermore, we plan to compare our work with other decomposition approaches and conduct more experiments to show the scalability of our method. Finally, we plan to integrate our work into a state-of-the-art model-based tool as a prototype.

References

1. Aleti, A., et al.: Archeopterix: An extendable tool for architecture optimization of aadl models. In: Proceedings of ICSE'09. pp. 61–71 (2009)
2. Aleti, A., et al.: Software architecture optimization methods: A systematic literature review. *IEEE Trans. Softw. Eng.* 39(5), 658–683 (May 2013)
3. AUTOSAR: Technical Overview v2.2.2. Tech. rep., AUTOSAR Development Cooperation (2011)
4. Blom, H., et al.: East-adl: An architecture description language for automotive software-intensive systems. *Embedded Computing Systems: Applications, Optimization, and Advanced Design* p. 456 (2013)
5. Bonfiatti, A., et al.: An efficient and complete approach for throughput-maximal sdf allocation and scheduling on multi-core platforms. In: DATE. pp. 897–902. IEEE (2010)
6. Bordeaux, L., et al.: Experiments with massively parallel constraint solving. In: IJCAI. vol. 2009, pp. 443–448 (2009)
7. Broy, M., Kirstan, S., Krömer, H., Schätz, B., Zimmermann, J.: What is the benefit of a model-based design of embedded software systems in the car industry? *Software Design and Development: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications* p. 310 (2013)
8. Hegedüs, Á., et al.: A model-driven framework for guided design space exploration. *Automated Software Engineering* 22(3), 399–436 (2015)
9. Jermann, C., et al.: Decomposition of geometric constraint systems: a survey. *International Journal of Computational Geometry & Applications* 16, 379–414 (2006)
10. Kang, E., et al.: An approach for effective design space exploration. In: Monterey Workshop. pp. 33–54. Springer (2010)
11. Manolios, P., et al.: Automating component-based system assembly. In: Proceedings of the 2007 international symposium on Software testing and analysis. pp. 61–72. ACM (2007)
12. Neema, S., et al.: Constraint-based design-space exploration and model synthesis. In: International Workshop on Embedded Software. pp. 290–305. Springer (2003)
13. Pohl, K., et al.: *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*. Springer Science & Business Media (2012)
14. Prakash, S., et al.: Sos: Synthesis of application-specific heterogeneous multiprocessor systems. *Journal of Parallel and Distributed computing* 16(4), 338–351 (1992)
15. Ruggiero, M., et al.: Communication-aware allocation and scheduling framework for stream-oriented multi-processor systems-on-chip. In: DATE Proceedings. pp. 3–8. European Design and Automation Association (2006)
16. Satish, N., et al.: A decomposition-based constraint optimization approach for statically scheduling task graphs with communication delays to multiprocessors. In: DATE Proceedings. pp. 57–62. EDA Consortium (2007)
17. Saxena, T., Karsai, G.: A meta-framework for design space exploration. In: (ECBS), 2011 Proceedings. pp. 71–80 (April 2011)
18. Sztipanovits, J., et al.: Design tool chain for cyber-physical systems: lessons learned. In: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC). pp. 1–6. IEEE (2015)
19. Voss, S., et al.: Deployment and scheduling synthesis for mixed-critical shared-memory applications. In: ECBS. pp. 100–109. IEEE (2013)
20. Yokoo, M., Durfee, E.H., Ishida, T., Kuwabara, K.: The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on knowledge and data engineering* 10(5), 673–685 (1998)