

# Design Space Exploration with Answer Set Programming

Christian Haubelt · Kai Neubauer · Torsten Schaub · Philipp Wanko

Received: date / Accepted: date

## 1 Project Overview

With increasing demands for functionality, performance, and energy consumption in both industrial and private environments, the development of corresponding embedded processing systems is becoming more and more intricate. Also, desired properties are conflicting and compromises have to be found from a vast number of options to decide the most viable design alternatives. Hence, effective Design Space Exploration (DSE; [8]) is imperative to create modern embedded systems with desirable properties; it aims at finding a representative set of optimal valid solutions to a design problem helping the designer to identify the best possible options.

The aim of our project *Design Space Exploration with Answer Set Programming* is to develop a general framework based on Answer Set Programming (ASP) that finds valid solutions to the system design problem and simultaneously performs DSE to find the most favorable alternatives. In the past, meta-heuristic algorithms were used almost exclusively, but they do not guarantee optimality and are ineffective for finding feasible designs in

highly constrained environments. ASP-based solutions alleviate this problem and have shown to be effective for system synthesis. Also, recent developments in ASP solving allow for a tight integration of background theories covering all (numeric) constraints occurring in DSE. This enables partial solution checking to quickly identify infeasible or suboptimal areas of the design space. Our project leverages these techniques to create a holistic framework for DSE using ASP modulo Theories.

## 2 Design Space Exploration

While DSE can be done at various abstraction levels, the overall goal is to identify one or several optimal implementations given a set of applications and a hardware platform. Our project targets streaming applications (such as video decoders) and heterogeneous hardware platforms organized as networks on chip (such as many-core SoCs) described at the electronic system level. Here, applications are defined as task-level descriptions and hardware platforms comprise networks of processing and memory elements. The DSE problem is twofold: first, evaluate a single feasible implementation, called a *design point*, and second, cover multiple (optimal) design points of the design space during exploration.

Obtaining a feasible implementation given a hardware platform and a set of applications is typically divided into three steps: *binding*, *routing*, and *scheduling*. Binding describes the process of allocating a resource for a specific task, routing ensures that messages of communicating tasks are correctly delivered through

---

Christian Haubelt  
University of Rostock E-mail: christian.haubelt@uni-rostock.de

Kai Neubauer  
University of Rostock E-mail: kai.neubauer@uni-rostock.de

Torsten Schaub  
University of Potsdam E-mail: torsten@cs.uni-potsdam.de

Philipp Wanko  
University of Potsdam E-mail: wanko@cs.uni-potsdam.de

the network, and scheduling assigns starting points for all tasks and communications so that no conflicts occur while executing applications.

By assigning worst-case execution times to tasks, as well as energy consumption and costs to resources, we are able to evaluate several quality criteria of a design point. We mainly focused on *latency*, i.e., a design point is better if all applications are finished in a shorter amount of time, *energy consumption* and *hardware cost*. These quality criteria are aggregated via a *Pareto* preference, i.e., a design point is better if it is at least as good in all criteria and strictly better in at least one when compared to another design point. Note that this preference might lead to a vast amount of optimal solutions since design points might be incomparable.

### 3 Design Space Exploration using Answer Set Programming Modulo Theories (ASPmT)

Until now, the project's focus lay on developing exact and flexible methods using ASP technology for finding design points for complex system models, obtaining optimal design points, and enumerating and storing optimal design points. This was achieved as a cooperation between our two groups at Rostock and Potsdam. While Rostock provided the expert domain knowledge, Potsdam developed the underlying technology.

In detail, in [6], we propose a novel ASPmT system synthesis approach. It supports more sophisticated system models, and makes use of tightly integrated background theories and partial solution checking. We present a comprehensive ASPmT encoding of all aspects of system synthesis, i.e., binding, routing, scheduling. As underlying technology, we use the ASP system *clingo* [2] whose grounding and solving components allow for incorporating application- or theory-specific reasoning into ASP. Furthermore, in [3], we instantiate the theory framework of *clingo* with linear constraints over reals and integers. The resulting systems *clingo[DL]*, *clingo[LP]* and *clingcon* handle the background theories of difference logic, integer linear programming and constraint programming, respectively. In [7], we present a novel approach to a holistic system level DSE based on ASPmT. DSE including feasibility check and optimization is performed directly within the solving process. To achieve that, we include additional background theories that concurrently guarantee compliance with hard constraints and perform the simultaneous optimization

of several design objectives. Binding, routing, scheduling and design objectives are represented in a declarative fashion in one encoding. Experimental results show the applicability of our approach for large optimization problems of up to 170 tasks mapped to 3-dimensional hardware platforms. Since there may be many optimal design points, further filtering and efficient storing of best known solutions is crucial. To present a smaller subset of representative solutions, we introduce a framework for computing diverse (or similar) solutions to logic programs with preferences in [9]. Furthermore, we explore efficient storage techniques for the best known solutions by utilizing quad-trees for saving design points during DSE in [5]. Finally, we address comparability of different DSE techniques in [4] by proposing a methodology for test case generation and presenting a versatile and easily expendable benchmark generator based on ASP that is able to produce hard synthesis problem instances.

### References

1. Carro, M., King, A. (eds.): Technical Communications of the Thirty-second International Conference on Logic Programming (ICLP'16), vol. 52. OASiCs (2016)
2. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with clingo 5. In: [1], pp. 2:1–2:15
3. Janhunen, T., Kaminski, R., Ostrowski, M., Schaub, T., Schellhorn, S., Wanko, P.: Clingo goes linear constraints over reals and integers. *Theory and Practice of Logic Programming* **17**(5-6), 872–888 (2017)
4. Neubauer, K., Haubelt, C., Wanko, P., Schaub, T.: Systematic test case instance generation for the assessment of system-level design space exploration approaches. In: Einundzwanzigster Workshop für Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV'18) (2018). To appear
5. Neubauer, K., Haubelt, C., Wanko, P., Schaub, T.: Utilizing quad-trees for efficient design space exploration with partial assignment evaluation. In: Proceedings of the Twenty-third Asia and South Pacific Design Automation Conference (ASP-DAC'18), pp. 434–439. IEEE (2018).
6. Neubauer, K., Wanko, P., Schaub, T., Haubelt, C.: Enhancing symbolic system synthesis through ASPmT with partial assignment evaluation. In: Proceedings of the Twentieth Conference on Design, Automation and Test in Europe (DATE'17), pp. 306–309. IEEE (2017)
7. Neubauer, K., Wanko, P., Schaub, T., Haubelt, C.: Exact multi-objective design space exploration using ASPmT. In: Proceedings of Design, Automation and Test in Europe (DATE'18) (2018). To appear
8. Pimentel, A.: Exploring exploration: A tutorial introduction to embedded systems design space exploration. *IEEE Design & Test* **34**(1), 77–90 (2017)
9. Romero, J., Schaub, T., Wanko, P.: Computing diverse optimal stable models. In: [1], pp. 3:1–3:14