Domain Polymorph Components For Explicit Opaque Borders in Hierarchical Heterogeneous Models *

Mohamed FEREDJ
University of Science and Technology Houari Boumediene
Faculty of Electronic and Computer Science
Department of Computer Science
32 Al-Alia, Beb Ezzouar,Algiers, Algeria
mohamed.feredj@gmail.com

Frédéric BOULANGER
Supélec - E3S, Department of Computer Science
Plateau de Moulon, 3 rue Joliot-Curie
91192 Gif-sur-Yvette
cedex, France
frederic.boulanger@supelec.fr

ABSTRACT

Hierarchical heterogeneous modeling is an approach for modeling heterogeneous embedded systems. It is a solution to manage the heterogeneity problem of the embedded systems and it helps to reduce the complexity of embedded systems modeling. However, the borders between hierarchical levels are opaque.

Domain polymorph components, which is another approach to heterogeneity, can automatically adapt their behavior to the semantics of the model in which they are used.

This article propose to use domain polymorph components at the border of the hierarchical levels of an heterogeneous model to make the semantic adaptation between these levels explicit.

1. INTRODUCTION

Embedded systems rely on several technologies. Thus, they are heterogeneous and very complex. They obey to a set of models of computation (MoCs). The MoCs correspond to the physical rules that govern the interactions of their subsystems.

The model of an embedded system must describe its properties and the constraints on its design. It must also allow its subsystems to be composed and to interact according to the MoCs that give their semantics. To allow the interactions between heterogeneous sub-models, the modeling of embedded systems must use heterogeneous approaches.

In a hierarchical heterogeneous approach, they are modeled using a set of models of computation (MoCs). The hierarchical approach [?] is an heterogeneous approach for managing heterogeneity which simplifies the modeling of the systems.

Although the hierarchical approach makes the manipulation of the heterogeneity of embedded systems easier and allows the combination of MoCs in a simple manner, the main drawback of this approach is the fact that each hierarchical level sees other hierarchical levels as black boxes. This makes the borders of hierarchical levels opaque (implicit).

In order to overcome this drawback, we propose to use domain polymorph components (DPCs) [?, ?] at the border of hierarchical levels to explicit the passage between MoCs. Our proposition improves the accuracy of the hierarchical heterogeneous approach.

*EWiLi'14, November 2014, Lisbon, Portugal.

2. THE HIERARCHICAL APPROACH

The hierarchical approach organizes the embedded system into hierarchical levels where each level contains subsystems that obey the same MoC. Therefore, using another MoC implies a change of level in the hierarchical organization. This approach is used by most heterogeneous modeling platforms for the design and simulation of embedded system and is considered as an efficient way of managing the complexity of such systems [?].

Figure 1 shows the hierarchical modeling of a typical heterogeneous embedded system that is composed of two subsystems A and B. The subsystem A obeys MoC M1 and the subsystem B obeys MoC M2. Thus, these subsystems are heterogeneous and so they cannot interact and communicate directly.

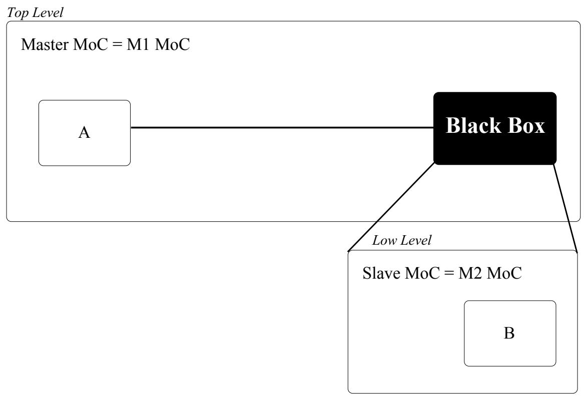

Figure 1: Hierarchical model of the heterogeneous system.

2.1 Drawbacks of the Hierarchical Approach

The main advantage of hierarchical approach is the abstraction of the sub-systems. However, the hierarchical approach has several drawbacks [?]. The main one, for which this article gives a solution, is the difficulty to answer the question: What happens at the border between hierarchical levels and how control, time and data cross the borders?

In order to answer this question, we propose to use domain polymorph components to make the borders of hierarchical levels explicit. Our contribution makes the designer's task easier, which increases the quality of the development cycle of embedded systems.

3. THE DOMAIN POLYMORPH COMPONENT – DPC –

A domain-polymorph component (DPC), see figure 2, is a component which has the capability to adapt its internal behavior (called core behavior) to the semantics of a host MoC. This adaptation is realized as follows: first, a DPC provides its internal core with an MoC that corresponds to the semantics its core behavior; second, the DPC is able to interpret control, time and data from its host MoC and to translate its outputs into the semantics of the host MoC [?].

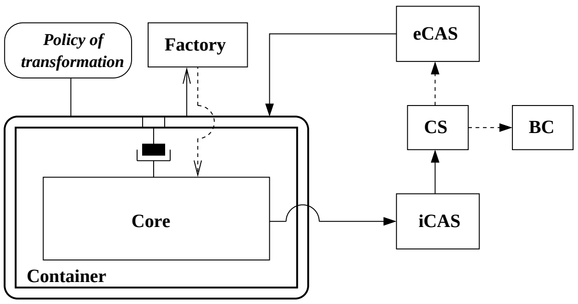

Figure 2: Architecture of the DPC.

4. USING DPC AT THE BORDERS OF HIERARCHICAL LEVELS

To make the adaptation of control, data and time from a hierarchical level to another explicit, we must model the semantic adaptation between the MoC of the outer level and the MoC of the inner level at the border between hierarchical levels. This is done using DPCs, because they have the ability to preserve semantics under any compatible MoC [?]. So, our proposition consists in placing DPCs at the border between hierarchical levels: one DPC at the outer level and another at the inner level. When building the model of the system, the DPC placed at the outer level is a complete DPC, i.e. it uses all its components (Container, Core, eCAS, iCAs, BR and CS), contrary to the DPC placed at the inner level which does not have a core component yet. The behavior of the Core describes how to adapt from the outer level to inner level and allows to designer to specify this adaptation in several ways, which is a part of the modeling process. Thus, the Core component is the gateway or bridge for semantic adaptation between MoCs.

When the model is executed, and after activation the DPC of the outer level, the Core component will be moved to the DPC of the inner level.

We modeled the system given on figure 3 to show how to use DPCs at the border of hierarchical levels, see figrue 3. This is the result of the combination of the hierarchical approach and DPCs.

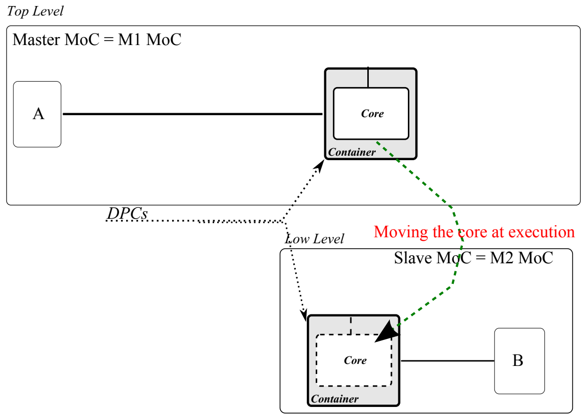

Figure 3: Use of DPCs at the border of hierarchical levels.

5. CONCLUSION

The hierarchical approach is an approach for managing heterogeneity that reduces the complexity of embedded systems modeling. Its strong point is the decomposition of a system into subsystems. However, the borders of hierarchical levels are opaque and the adaptation of data, time and control between hierarchical levels is implicit. Consequently, the designer cannot know what happens at the borders and so the designer cannot describe the adaptation between hierarchical levels.

In this paper, we proposed to use Domain Polymorph Components at the border of hierarchical levels to allow the designer of an embedded system to define explicitly how the adaptation of semantics between hierarchical levels can be done?

6. REFERENCES

[1] J. Eker, J. W. Jannek, and al. Timing heterogeneity – the ptolemy approach. In *IEEE, special issue on modeling and design of embedded software*, pages 127–144, January 2003.

[2] M. Feredj, F. Boulanger, and A. M. Mbobi. A model of domain-polymorph component for heterogeneous system design. *Journal of Systems and Software*, 82(1):112–120, 2009.

[3] M. Feredj, F. Boulanger, and M. Mbobi. An approach for domain polymorph components design. In *IEEE International Conference on Information Reuse and Integration*, pages 127–144, November 2004.

[4] A. Girault, B. Lee, and E. A. Lee. Hierarchical finite state machines with multiple concurrency models. *IEEE Transactions On Computer-aided Design Of Integrated Circuits And Systems*, 18(6):112–120, June 1999.

[5] A. Goderis, C. Brooks, I. Altintas, E. A. Lee, and C. Gobel. Heterogeneous composition of models of computation. *Future Generation Computer Systems.*, 25(5):552–560, May 2009.