UML/MARTE METHODOLOGY FOR AUTOMATIC SYSTEMC CODE GENERATION OF OPENMAX MULTIMEDIA APPLICATIONS

Pablo Peñil, Pablo Sánchez
University of Cantabria
Santander, Spain
{pablop, Sanchez}@teisa.unican.es

David de la Fuente, Jesús Barba, Juan Carlos López
University of Castilla La Mancha
Albacete, Spain
{David.Fuente, Jesus.Barba, JuanCarlos.Lopez}
@uclm.es

ABSTRACT

The design of multimedia systems is becoming more and more challenging due to their growing complexity and strict performance requirements. New implementations fulfilling new market requirements must be included in increasingly complex multimedia systems. In this context, the OpenMAX standard provides abstraction facilities needed in order to design multimedia streaming applications with large amounts of data processing.

This paper presents a methodology which provides a UML/MARTE methodology for the specification of multimedia systems according to the OpenMAX specification requirements for system level-design activities and a framework which automates the extraction of the corresponding SystemC executable specification, enabling the execution and the verification of the multimedia system.

Index Terms— *UML/MARTE, UTP, SystemC, OpenMAX, automatic generation*

1. INTRODUCTION

The design of embedded systems is in a highly competitive context. The translation of a successful design into a successful product depends greatly on becoming the first product on the market with new complex functionalities and strict performance constraints. More specifically, in the multimedia system design area, new devices include a wide set of applications with an increasing number of intensive data processing operations fulfilling the new standards of audio, video, image quality.

To cope with this increment of system requirements, some parts of the system have been implemented in electronic devices, making HW/SW integration necessary. The lack of standardization in HW/SW integration has led to ad-hoc implementations whereas the adoption of standards provides portable, flexible and reusable designs for embedded systems.

The OpenMAX standard [19] is an initiative promoted by the Khronos Group that defines a standardized media component interface to enable developers and platform providers to integrate and communicate with multimedia codecs implemented in hardware or software.

A strategy for designing complex systems has been the development of electronic system-level (ESL) design methodologies [13], where the key initial activity is specification. Model-driven development (MDD) methodologies enable concepts for making specifications simpler and more understandable, which are major requirements for tackling the design challenge [14]. Using standard languages, such as UML [15], provides understandability and portability of specifications.

Model-driven design methodologies are commonly adopted to handle the design of large functionalities. The latest design methodologies start from high-level UML models combined with algorithmic codes (e.g. C, C++, Matlab, etc.) of the different system components [15]. In these models, the user defines the system functionality using a platform-independent model (PIM). Then, given a platform definition model (PDM), the PIM is translated to one or more platform-specific models (PSMs), where resource allocations are specified.

In order to provide specific semantics for UML, profiles have been developed. In the specific context of embedded systems, the MARTE profile [18] has been developed to model and analyze real-time systems, providing the concepts needed to describe real-time features that specify the semantics of these kinds of systems at high abstraction levels. The UML Testing Profile [17] enables the definition of models that capture scenarios for system testing.

In order to achieve an optimal solution, different design alternatives have to be considered. Design space exploration (DSE) solutions have been proposed to perform this selection process. These DSE solutions should be captured in the UML/MARTE model, specifying all the different design alternatives that can be considered, defining a set of test benches which enables different system executions in order to obtain performance results to obtain the final system design. Count on the help of a system model in an early stage of development can facilitate the evaluation of the different design alternatives, reducing development costs. For this reason, SystemC [25] has been used in this work, since it is a modeling language that is applied to system-level modeling, architectural exploration, performance modeling, software development, functional verification, and high-level synthesis.

This paper focuses on the description of a design flow that enables the specification of a multimedia system design under the OpenMAX standard. By using the modelling capabilities of UML and the standard profiles, MARTE and UTP, the designer can completely specify the most relevant, requisite multimedia system attributes needed to obtain, in an automatic way, the SystemC executable specification that corresponds to a *HW OpenMAX Integration Layer* infrastructure [20]. This HW OpenMAX IL infrastructure can be executed and explored, considering different automatically generated test-benches as well.

The paper is organized as follows; in section 2, a study of the state-of-the-art is presented. In section 3 the complete proposed design flow is described. Finally, some conclusions and future work are presented in section 4.

2. STATE-OF-THE-ART

Through the use of OpenMAX in embedded systems, developers can reduce the effort required to design multimedia hardware because: (a) the whole core logic can be reused when targeting a new platform (standardized interfaces), (b) it is not necessary to hand write the drivers or code that depends on these components and, (c) communication issues are separated from the processing primitives (the same synchronization protocols, standardized communication mechanism, etc.).

The implantation of OpenMAX in many commercial products is a fact. For example, in [21], NVIDIA offered a demonstration which showed a prototype OpenMAX IL implementation executing on an NVIDIA GeForce 3D handheld graphics processing unit (GPU) to create a flexible, accelerated streaming media pipeline. Later, NVIDIA developed the first dual-core processor for faster Web browsing and snappier response time called NVIDIA Tegra 2 [22], which supports OpenMAX for media acceleration. The GST-OpenMAX project [23] extends the GStreamer multimedia framework with OMXIL in order to provide the advantage of enabling access to multimedia components in a standardized way. Many component wrappers have been developed for different OMXIL implementations, such as Bellagio, TI (OMAP-3430) or Maemo. GST-OpenMAX is already distributed in many embedded platforms, such as Angstrom and Maemo.

The application scope of UML has been extended to cover different domains to its initial application domain, such as object-oriented software system modelling. In this context, in [6], the capabilities of the application for the design of electronic systems are described. Specifically, in order to exploit the benefits of UML as a modelling language, the MARTE profile was created to deal with the modelling of real-time embedded systems.

In [7], a UML/MARTE methodology for specifying systems with heterogeneous communication semantics is presented. Then, a mapping to SystemC is described in order to obtain an executable specification. Gaspard2 [9] is a design environment for data-intensive applications which enables TLM SystemC platforms to be generated from MARTE descriptions of both the application and the hardware.

Other works take UML/MARTE models as input and generate executable code from them. In [8] the complete design flow to move from high-level MARTE models to code generation, for implementation of dynamically reconfigurable SoCs, is presented. In this paper, generic control semantics for the specification of adaptive and dynamic reconfigurable SoCs is presented. In [24], a component-based modelling methodology based on UML/MARTE and explicitly designed for supporting design space exploration is presented. This UML/MARTE design exploration methodology is focused on the specification of a set of design solutions for the HW/SW architecture; characteristics of the HW resources (frequency, etc.) and multiple application allocations to HW resources.

According to the previous works, the use of UML/MARTE as high-level language for embedded systems is widely extended. Additionally, the generation of SystemC executable specifications from UML models is a well-known activity. However, no model-based methodology has been found that specify a complete design flow that based basing on UML/MARTE high-level specification, enables the automatic SystemC code generation considering the standard OpenMAX for dealing with the current multimedia system in a feasible way.

3. PROPOSED FLOW

The proposed design flow (Figure 1) starts from the creation of a UML/MARTE model.

The system modelling methodology described in this paper is characterized by following a component-oriented approach [3] and applying Model-Driven Architecture (MDA) [1] principles in the development of the HW/SW embedded systems. Moreover, the proposed approach makes this methodology software centric [2] as it considers application components as units allocable either in the software resources or hardware resources.

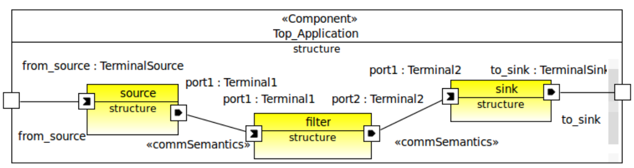

Figure 1 Application structure

The information contained in a UML/MARTE model is separated in specific concerns, depending on their application area. Each concern is captured in a model view, which is represented using the UML diagrams that best fit the concern. These model views define the data types, the application components, the interfaces for the communication of these application components, the application structure (Figure 1), the HW/SW elements used for defined the platform where the application components are mapped, etc. Additionally, the views of the system model are grouped according to the MDA guide, forming three different viewpoints: the Platform-Independent Model, PIM), the Platform Description Model, (PDM), and Platform-Specific Model (PSM). The PIM (Figure 1) describes the system functionality (e.g. application, functional code, interfaces, application structure, communication mechanism…). The PDM describes the different HW and SW resources that form part of the system platform. Finally, the PSM describes the system platform architecture and the allocation of the application components into the platform resources.

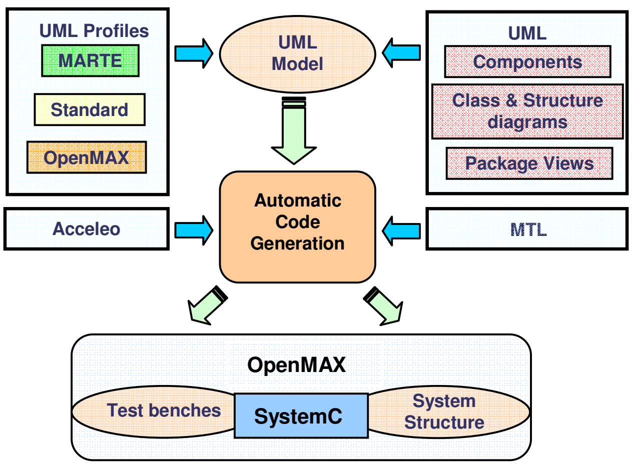

Figure 2 Proposed OpenMAX Flow

From this model, an infrastructure integrated in Eclipse generates all the OpenMAX-SystemC code required for simulation or execution. The graphical tool used to create the UML/MARTE model is Papyrus [10]. A code generator will be developed as a set of generation templates written in the standard *MTL* language [12]. The development will be done through *Acceleo* [11], a code generation framework fully integrated in *Eclipse*.

From the UML/MARTE model of the system to be designed, the goal is to automatically generate a SystemC specification of the system in order to get the ability to know its temporal behavior considering the different design configurations. This SystemC specification is executed with a set of test benches in order to select the most adequate configuration parameters of the system components.

According to the characteristics of the implementation of each OpenMAX component (described in [20]) modelled in the UML/MARTE model, the code generation produces a SystemC component specification (Figure 3).

The functionality of the component is translated into the Media Core (MC). Internally, the MC contains a generic synchronous pipeline in which the number of stages and cycle time can be set, making the MC temporal behavior flexible and configurable. In order to make computation and communication independent, a placeholder for this MC, called OpenMAX Adapter (OA), has been implemented and is responsible for interpreting the OpenMAX primitives. Both elements form the OpenMAX Component.

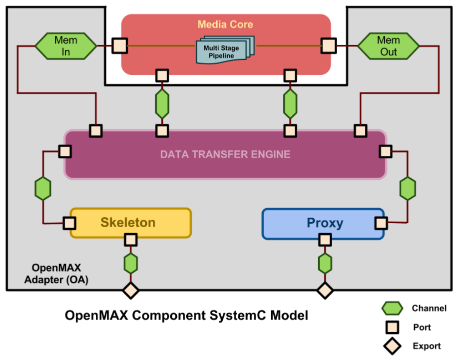

Figure 3 SystemC OpenMAX Component specification

A simple fixed interface is provided between the MC and the adapter, mainly based on a standardized memory interface. Therefore, HMCs remain independent from the data communication protocols and the memory technologies. This interface facilitates reading and writing buffers coming from MC to local memories present in the OA. As it is defined in the standard, a buffer is the minimum amount of media data that can be exchanged between two OpenMAX components.

Besides the memory interface, the OHA (OpenMax Hardware Adapter) implements the control logic that governs the MC execution. Moreover, the MC provides information to the OA about its execution status that is interpreted by the Data Transfer Engine (DTE) to overlap MC execution with the data transmission process that can be carried out through the Proxy and Skeleton communication adapters (acting as OpenMAX ports). Using the rules for modeling in SystemC, all modules in the OpenMAX Component are connected via ports and channels as described in Figure 4.

In order to check the performance of the set of interconnected OpenMAX Components, a simulation scenario has been developed. In it, several simulations will be launched according to the value of each component configuration parameter. From a pool of test cases, each simulation reads one test case in which the number of OpenMAX Components in the chain, the amount of media data that will be processed and the configuration parameters for each component are defined.

The simulation process is made up of a chain of OpenMAX Components connected through a shared bus (specifically through an AHB bus SystemC model as can be seen in Figure 4) and the simulation starts by injecting the media data into the chain. These data will go through the components and will be collected at the end of the chain. At this point, the time taken to complete the process is recorded.

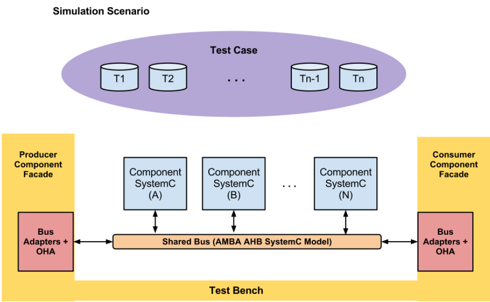

Figure 4 SystemC Simulation Scenario

Depending on the results obtained after the simulation of all test cases, the configuration of each component is selected. The selection criterion is easy, the configuration parameters of the simulation whose execution time has been the shortest are chosen. In this kind of application, a typical performance metric is the execution time, so less time indicates better performance.

4. CONCLUSIONS AND FUTURE WORK

The paper presents a UML/MARTE based design flow in order to specify of current multimedia systems according to the specification requirements that the standard OpenMAX provides. In addition, the high-level modeling methodology enables the specification of test-benches which are used to establish a design exploration process in order to find the best configuration of the system.

Then, from the UML/MARTE model, a SystemC mapping is produced. This SystemC mapping enables the automatic code generation of a SystemC executable specification in order to obtain the different timing execution performances. Depending on the values obtained, the best system configuration can be selected.

From the on-going work presented in this paper, the code generator will be implemented. This code generator will be implemented by using MTL. Then, it will be included in an Eclipse infrastructure, which will enable the application of the UML/MARTE OpenMAX methodology in the complete design of multimedia systems.

4. ACKNOWLEDGMENTS

This research was supported by the Spanish Ministry of Science and Innovation under the project DREAMS (TEC2011-28666-C04-03), and by the Spanish Ministry of Industry and the Centre for the Development of Industrial Technology under the project ENERGOS (CEN-20091048).

8. REFERENCES

[1] D. C. Schmidt, "Model-driven Engineering" IEEE Computer, vol. 39 no. 2, pp. 25-31, 2006.

[2] K. Yamashita. "Possibility of ESL: A software centric system design for multicore SoC in the upstream phase", Design Automation Conference (ASP-DAC), 2010.

[3] C. Szyperski, Component Software: Beyond Object-Oriented Programming. 2nd ed. Addison-Wesley Professional, 2002.

[4] OMG: "MOF Model To Text Transformation Language (MOFM2T), 1.0", http://www.omg.org/spec/MOFM2T/1.0/, 2008. OMG. MOF Model To Text Language. Jan., 2008.

[5] Website. www.acceleo.org. Nov., 2010.

[6] Y. Vanderperren, W. Mueller, W. Dehaene. UML for Electronic Systems Design: a comprehensive overview. Journal on Design Automation for Embedded Systems, Springer Verlag, August 2008.

[7] P. Peñil, J. Medina, H. Posadas, E.Villar. Generating Heterogeneous Executable Specifications in SystemC from UML/MARTE Models. UML-FM December, 2009.

[8] I. R. Quadri, H. Yu, A. Gamatié, E. Rutten, S. Meftali, J-L. Dekeyser. Targeting reconfigurable FPGA based SoCs using the UML MARTE profile: From high abstraction levels to code generation. International Journal of Embedded Systems, 2010.

[9] É. Piel, R. Atitallah, P. Marquet, S. Meftali, S. Niar, A. Etien, J.-L. Dekeyser, P. Boulet: "Gaspard2: from MARTE to SystemC Simulation", proc. of the DATE'08 workshop on Modeling and Analysis of Real-Time and Embedded Systems with the MARTE UML profile, 2008.

[10] http://www.papyrusuml.org/

[11] Website. www.acceleo.org. Nov., 2010.

[12] OMG. MOF Model To Text Language. Jan., 2008.

[13]G. Martin, B.Bailey, A. Piziali. ESL Design and Verification: A Prescription for Electronic System Level Methodology (Systems on Silicon). March 9, 2007. ISBN-10: 0123735513.

[14]H. Kopetz. The Complexity Challenge in Embedded System Design. In 11th IEEE ISORC.

[15] Y. Vanderperren, W. Mueller, and W. Dehaene, "UML for electronic systems design: a comprehensive overview," Design Automation for Embedded Systems, vol. 12, no. 4, 2008

[16] http://www.omg.org/spec/UML/2.4/. February, 2013.

[17] OMG. UML Testing Profile (UTP) 1.1. http://utp.omg.org/.

[18] OMG. MARTE Profile 1.1. http://www.omgmarte.org/

[19] D. de la Fuente, J. Barba, F. Rincón, J. D. Dondo and J.C. López. Embedded Systems - High Performance Systems. ISBN: 978-953-51-0350-9. Pag 129-154

[20] NVIDIA Demonstrates High Definition Processor. Las Vegas, Nevada. January 4, 2006

[21] NVIDIA Khronos Apps SDK, 2010. http://www.nvidia.com

[22] Adaptive Digital Technologies, Inc. Adaptive Digital OpenMAX IL Implementation. 2012. http://www.adaptativedigital.com

[23] VisualOn. Enabling Home Entertainment and Mobile Multimedia. 2011 TI Technology Day. Taipei. June 16.

[24] F. Herrera, P. Peñil, E. Villar, F. Ferrero and R. Valencia "An Embedded System Modelling Methodology for Design Space Exploration". JCE 2012.

[25] SystemC website. http://www.accellera.org/