Using MARTE in a Co-Design Methodology

Ali Koudri  
Thales Aerospace Division  
ali.koudri@fr.thalesgroup.com

Denis Aulagnier  
Thales Aerospace Division  
denis.aulagnier@fr.thalesgroup.com

Didier Vojtisek  
IRISA  
didier.vojtisek@irisa.fr

Philippe Soulard  
Sodius  
psoulard@sodius.com

Christophe Moy  
Supelec  
christophe.moy@supelec.fr

Joël Champeau  
ENSIETA  
joel.champeau@ensiesta.fr

Jorgiano Vidal  
Lab-STICC  
jorgiano.vidal@univ-ubs.fr

Jean-Christophe Le Lann  
Thomson  
jean-christophe.le-lann@thomson.net

Abstract—The Model Driven Architecture is a promising approach aiming to fill the productivity gap due to the increasing technology and time to market pressure. In the field of real time embedded systems, this approach requires the use of well-adapted formalisms in a reliable process that guarantees the quality of the products. MARTE, the new standardized UML profile, provides those formalisms that are applied in the process defined in the MoPCoM SoC/SoPC research program, aiming to develop SoC and SoPC applications. In this paper, we discuss the main phases of this process and their use of MARTE.

I. INTRODUCTION

Nowadays, embedded equipments rely more and more on components such as System on Chip (SoC) or System on Programmable Components (SoPC) based on Field Programmable Gate Array (FPGA) for low volume of production.

Embedded applications require more and more computing resources. These needs can be fulfilled by SoC / SoPC integrating on the same component analog circuits, hardwired ASIC Gates, programmable logical cells, memory blocks, high speed links, DSP and processor cores. Nevertheless, the SoC/SoPC development requires challenging productivity improvement to be able to address the component design complexity[ITR07].

Moreover, some embedded applications are constrained by certification norms like ARP-4754 (system), DO-178B (software) and DO-254 (hardware). In such case, SoC/SoPC developments have to be handled by rigorous methodologies based on well-adapted formalisms[KCA07].

In this paper, we discuss a new methodology to develop SoC/SoPC applications. This methodology is based on UML and MDA (Model Driven Architecture), and takes into account the achievements of co-design community. MDA[OMG03], promoted by the OMG (Object Management Group) replaces classical waterfall process (specification, design, coding), by a succession of models transformations, among these models:

- CIM (Computation Independent Model) models the requirements for the system and how the the system will be used,
- PIM (Platform Independent Model) is the view of the application independent from the target platform and can be considered as a functional description,
- PM (Platform Model) represents the target architecture used to implement the application,
- PSM (Platform Specific Model) results from the mapping of the PIM on the PM.

Actually, several PM of the SoC/SoPC Platform with different levels of abstraction are required to achieve the final implementation. For each PM, the process of allocation results in new PSM and analysis models. This approach enables to increase flexibility, reuse and separation of concerns.

The MDA approach was initially proposed in the context of software development and can be applied in the co-design domain. Indeed, "MDA notions" have been introduced long time ago in this field. For example, the Y-Chart methodology proposed by Gajski and Kuhn identifies behavioural, structural and geometric viewpoints[DR83].

Compared to conventional co-design methodologies, the use of UML/MDA for SoC/SoPC development enables to take advantage of the standards and the tooling developed by the UML/MDA community, such as the new UML profile for Modeling and Analysis of Real Time Embedded Systems (MARTE)[OMG07]. This profile enriches UML adding new notions for real time systems such as Non-Functional Properties or Time Access. Those features are seen in the section III.

In this paper, we present a new Co-Design methodology defined in the MOPCOM research project, aiming to provide a full MDA compliant tool to design SoC/SoPC applications. In this methodology, system analysis follows the Telelogic Harmony System Engineering process, based on the use of SysML[OMG06], improved with the use of MARTE. It is then refined to address platform and allocation modeling. Three levels of abstraction have been defined to describe the platform on which the application is mapped (see fig.1):

- APA (Abstract Platform Architecture) considering the platform as a set of basic services without notion of time,
- EPA (Execution Platform Architecture) refining the APA model taking into account software and hardware nature of APA components and timing constraints,
- DPA (Detailed Platform Architecture) detailing the EPA model, taking into account other constraints (power consumption, surface, etc).

In order to favor analysis, reuse or separation of concerns (computation / communication), the description of the com-
munications are described following the TML 2 standard as defined by the OSCI [Ini07]. This allows the same design space exploration allowed by different TLM levels (PV, PVT, CC) and the separation of concerns allowed by the MDA approach. The Programmer View (PV) level captures the SoC/SoPC architecture without clock or explicit timing. It focuses on functional correction based on the execution of the underlying Model of Computation. Timing concern is taken into account in the Programmer View + Time (PVT) level where data size and computation duration are described following the non functional requirements. Only the Cycle Callable (CC) level allows the fine verification of all those properties.

At each level, several analysis models (performance, power consumption, etc.) can be generated from each allocation in order to validate design choices. This approach enables early errors detection and design space exploration with shorter simulation time. Moreover, in order to add meaningful properties to our methodology (predictability, maintainability, improvability, etc.) and automate it, we have used the SPEM profile[OMG05] to formalize it.

In this paper, we present our use of the UML MARTE profile through each phase of our methodology and we discuss the encountered problems from implementation or conceptual point of view.

II. RELATED WORKS

The relevance of model based approaches in Co-Design have been widely discussed by the literature. In [BDH06], authors enumerate the main advantages of such approaches: cost decrease, silicon complexity handling, productivity increase, etc. Although several methodologies based on the use of UML/MDA have been applied, UML were not tailored to design SoC/SoPC applications. Then, it was extended and toled by each of those methodologies, thanks to the profiling mechanism provided since UML 1.3. For example, in [BDDM04], authors propose extensions to describe Intensive Signal Processing Applications in a profile called ISP UML Profile. In [RSRB05], authors apply a SoC Design Methodology based on the use of a UML profile called UML for SystemC Profile. This profile allows direct mapping between the models and the concepts of the SystemC language. The same kind of approach is presented in [WZZ+06]. A SystemC UML profile is used to generate a TLM/PV SystemC code skeleton. In [CSL+], a design environment called Metropolis and aiming to design embedded systems is proposed. It tailors UML in order to describe execution platforms and their refinements. They provide set of stereotypes representing structural parts of the platform and corresponding model of computation.

In [RSRB07] a development process for embedded systems
It is interesting to notice how, through all those approaches, common needs like platform and allocation description at several levels of abstraction are fulfilled by different approaches. Some of them propose a direct mapping of the platform concepts to the target languages while others use target languages as concrete syntax in purpose of generation or execution. The problem with those approaches are the dependency to the tools implementing very specific profiles or DSL (Domain Specific Languages). With the standardization of the MARTE profile, we think this drawback will be handled by the use of common formalisms to describe real time and embedded systems.

### III. Use of MARTE in the MoPCoM SoC/SoPC research project

The MoPCoM process is a Co-Design Process based on the use of models and transformations. Since one of the main features of the MDA is the separation of concerns, there is a need to dispose of well-adapted formalisms to describe each part of the MDA model. For several reasons, such as communication improvements or tooling, we have made the choice to use standardized meta-models and profiles. The new standardized UML MARTE Profile add semantics to UML for Real time Embedded System Modeling. This profile is organized around several packages capturing different concerns:

- The Non Functional Properties (NFP) package enables to describe features that are not directly related to the business model (performance, memory usage, power consumption, etc.) and mechanisms to attach them to model elements,
- The Time package enables modeling of time structure and access (continuous time, clock, etc.),
- The Generic Resource Modeling (GRM) package enables modeling platform resources from high level perspective and mechanisms to manage access to those resources,
- The Allocation package enables modeling of spatial and temporal allocations,
- The Detailed Resource Modeling (DRM) package allows modeling of hardware and software resources at several levels of granularity; it contains two sub-packages enabling the modeling of Software Resource (SRM) and Hardware Resource (HRM),
- The Generic Quantitative Analysis Modeling (GQAM) package allows several types; It contains two sub-packages for modeling Performance Analysis (PAM) and Schedulability Analysis (SAM).

Table I sum up the requirements of the process and formalisms that satisfy those requirements. The system properties, measurable or not, like power consumption or latency, can be expressed using the Value Specification Language (VSL) provided by MARTE at each stage of the process.

### A. System Analysis

The system analysis goes from the requirements analysis to the system design and is covered by the SysML profile [OMG06]. Since this part is not really in the scope of this paper, we will just give a quick overview in the following points:

- requirements elicitation,
- static validation of the requirements,
- manual transformation of the requirements into use cases and interactions,
- use cases validation through simulation,
- identification of the subsystems,
- operational contracts allocation,
- interaction refinements,
- system design validation.

During this stage, we use some concepts of the NFP packages in order to describe the required quality of services in terms of real-time features of behaviors and interactions. At the end of this stage, an annotated functional architecture is provided for following steps, including allocations and refinements.

### B. Abstract Platform Architecture

The platform is defined as a set of resources like processing resources, communication media or storage resources. The purpose of this level is to describe applied mechanisms in order to realize high level functions. Communications between APA blocks are point-to-point and realized through abstract channels providing set of basic services allowing data transport. Actually, what we aim to describe at this level is the underlying model of computation supporting the execution of the application and since we just want to demonstrate the functional correction of this model, it is thus untimed. Synchronization mechanisms ensure the characterization of the causal relations between processing elements and then ensure the predictability of the system behavior. The fact that the model is untimed doesn’t mean that timing constraints inherited from the application model are not taken into account. Functional delays can be then inserted into the APA model, regarding the functional constraints.

---

1. Unified Process for Embedded Systems

<table>
<thead>
<tr>
<th>MoPCoM Process Requirements</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Computational Independent Model</td>
<td>SysML + MARTE NFP and RTE-MoCC</td>
</tr>
<tr>
<td>Model Platform Independent Model</td>
<td>SysML + MARTE NFP and RTE-MoCC</td>
</tr>
<tr>
<td>Model Abstract Platform Architecture</td>
<td>MARTE GRM and NFP</td>
</tr>
<tr>
<td>Model Execution Platform Architecture</td>
<td>MARTE DRM + NFP + Time</td>
</tr>
<tr>
<td>Model Detailed Platform Architecture</td>
<td>MARTE DRM + NFP + Time</td>
</tr>
<tr>
<td>Model Allocation (PIM to APA/EPA/DPA)</td>
<td>MARTE Allocation + NFP</td>
</tr>
<tr>
<td>Model Analysis Models</td>
<td>MARTE PAM and SAM</td>
</tr>
</tbody>
</table>

**TABLE I MoPCoM PROCESS REQUIREMENTS**
Once the platform has been defined by the software or hardware engineering teams, allocation can be modelled using well-adapted formalisms provided by the allocation package of MARTE. The allocation can be of spatial, temporal or hybrid nature. Since many allocations with different characteristics (performance or schedulability) are possible, models of allocation are concerned with the functional correctness issue. The figure 2 summarizes issues related to the allocation at the APA level. The application is modelled through business components communicating through (a)synchronous operation calls or events send/reception while the platform is modelled through APA blocks communicating through channels implementing data transport services. The execution semantic described in the business model must be verified through the allocation. Since the APA platform is untimed, there is no notion of protocol at this level.

![Fig. 2. APA allocation](image)

In order to describe the APA platform, one need to describe resources from a high level perspective, without prejudging of their hardware or software nature, and take into account their quality of service. The GRM package of MARTE provides all needed stereotypes and tagged values to represent resources (computing resources, storage resource, communication media, etc.) and their use. This package is used in conjunction with the NFP package for the quality of services and Time Modeling package for timing constraints. The GRM package provides also mechanisms of synchronization and resource management that fulfills the needs for APA modeling. The package Allocation Modeling provides all needed stereotypes to allocate business model to APA platform. Since an allocation comprises at least one application end and one execution platform end, we need to constraint the set of possible mappings for this level refining the allocation definition.

C. Execution Platform Architecture

The MOPCOM EPA platform level aims at providing a more precise platform than the APA level: while APA focused on point-to-point communication, EPA provides more details with respect to the expected final topology on the system. The right methodological tools at this level are clearly MARTE sub-profile HRM – Hardware Resource Modeling on which MOPCOM relies. Concerning the usage of MARTE, the stereotypes used here are hwComputingResource (processor,...), hwCommunication( bus,...).

EPA is geared towards architects willing to depict global scenario of data movement over the chip and computing resources, without committing to final IP choice. EPA is thus fundamental to preserve the representativity of application with respect to the platform, while enabling fast design space exploration and high-speed simulation, based on generic high-level components.

In this sense, many hardware resources must be adequately identified at this level, without over-specifying them, which is often the case when architects are not provided with appropriate tools to help their specification. A natural companion modeling concept of EPA is transaction-level modeling, as promoted by Gajski and SystemC community in general.

Among those resources, communication media like buses, NoCs and buffering medium need to appear in EPA (the right stereotypes here are HWI/O and HWStorage). Beyond the communication aspects, computing resources are also included in this model, possibly still in an abstract (behavioral) version. For instance, instead of describing a real RISC processor or accelerator, EPA proposes to alleviate the need for deep hardware description and invites to keep the functions executed by these processing elements as the representative of the blocks themselves.

Thus a wrapper needs to ensure that the function can indeed have access to the communication media: here we prone a simplified version of generic DMA at the I/O of the functional data-intensive blocks, possibly assisted with control/status registers that ensure the high-level control of the application in a processor-like manner. Using this approach, it is possible to set up dummy experiments to test these notions and their adequacy to a broader set of other experiments. In the following drawing, we depict typical experiment within reach of an EPA model: the goal is to move data from DDR zones to other DDR zones, after some modifications executed on the set of data. A global controller, mimicking a microcontroller/processor, observes the completion of the various data transfers in order to enable the next transfers. Entire data structures can be moved from one location to another without any serialization mechanism to low level data representation. As a consequence of this separation of concerns, early application porting debug activity can start, while studying the underlying lower level mechanisms constraints that need to be fulfilled to respect the expected final behavior.

Clearly here, the EPA level envisions the mapping process as a simple binding of application functions to processing elements and point-to-point communication links to concrete communication elements, with no further notion of compilation to dedicated binary format, nor the resort to ISS/interpreters: functions can still be seen as abstract tasks, just as in the APA level. Note that some blocks or units may be in charge of executing several functions: in this case, this coarse grain resource sharing then assumes the presence of a basic abstract arbitration mechanism.

The performances of the final hardware cannot be mea-
Fig. 3. EPA typical experiment: data are moved from one location to another, through well identified communication links, possibly simulated at a behavioral level, abstracting away details about finer details.

As emphasized here, EPA prones to resort to behavioral descriptions instead of deeply detailed descriptions. For instance, concerning the processors, the further refinement to real binary code (on the embedded SW side) and cycle & pin accurate hardware will be treated in the ’next’ level, i.e DPA level, where measures becomes meaningful. The same applied for communications, where the complete protocols will need to be fully modeled.

One interesting question is about the minimum set of hardware features to embed in this EPA level so that the model of computation can be implemented either manually or handled by the MDA tool-flow: register control, DMA control etc.

D. Detailed Platform Architecture

The DPA level aims to provide all required information to enable RTL code generation. The platform is defined as a set of IPs communicating through buses and all transactions are implemented using refined and precise protocols.

The DPA level refines the EPA level as follow:
- Introduction of hardware clock(s) – all synchronous elements must be connected to a clock,
- Definition of a cycle accurate communication protocol – all communications must be well defined in terms of signals and clock cycle,
- Detail of Physical hardware Layout – for example: card layout and power consumption.

Although the MARTE profile provides a set of stereotypes enabling to define buses and behavior, it does not enable an explicit way to precise protocols. The constraints related to protocols are expressed using NFP annotations. In order to faster the design and favor reuse, one can use libraries of standard protocol and generate needed wrappers. The MoPCoM project aims at providing such libraries.

The figure 4 shows the main issues related to the allocation at the DPA level. As the execution platform is detailed, the application must be refined to precise cycle accuracy of the behavior. Designers must then provide complementary views of the platform showing their physical details such as processor type, ISS, registers or pins. This can be achieved using stereotypes provided by the HRM MARTE sub-profile.

The consistency of the behaviors and the communications must be kept through the refinement. Indeed, the model of computation defined earlier is detailed with information related to time or data. Then, the desired performance can be checked at this level.

The last refinement to achieve an optimal configuration of the DPA blocks (physical characteristics) must be provided at this stage and the HRM sub-profile contains a package providing all needed stereotypes to achieve this goal. They are classified in two types:
- HwLayout: Channel, Card, Channel or Port,
- HwPower: Power Supply, Cooling Supply or Battery.

For instance, several tags are associated to the HwPort stereotype such as pin number, area, weight, etc. Then, allocation of the functional model to this very refined hardware...
platform definition enables the verification and validation of the non-functional properties required by the functional model as quality of service.

Several modeling constraints must be applied in order to allow an automatic code generation. Among them:

- HRM stereotypes must be used to model all hardware elements and all required tagged values must be set,
- Protocols must be specified using state-charts,
- Every synchronous element must be connected to a clock.

The model provided at the end of this stage is transformed into a RTL model for code generation. The target meta-model and code generation rules are provided by the Sodius Company in their MDWorkbench Tool.

IV. DISCUSSIONS

The development methodology we have presented above is being experimented on a video application for SoC and a RF digital receiver application for SoPC. Despite we think that MARTE can fulfill most of the requirements of a modelling language for SoC/SoPC design and analysis, we faced some limitations applying MARTE. Those limitations are either related to the implementation of the profile or missing concepts in the profile.

- **Allocation**: MARTE proposes to specify the "allocate" relationship between business components and resource components with notation based on a UML dependency (with "allocation" stereotype). This graphical approach can be very cumbersome and messy for large-models. Some tooling, using widgets for example, could be developed to facilitate the allocation activity.

- **Model of Computation and Communication**: MARTE foundation enables to define alternative MoCC. Modelling or implementation of embedded applications generally rely on heterogeneous MoCCs such as Synchronous Data Flow (SDF), Communication sequential process (CSP), Petri Net (PN), Kahn Process Network (KPN). These specific MoCCs should be introduced in MARTE.

- **Action Semantics**: In the purpose of executing and testing models, UML has introduced the Action Semantics to provide software independent language specification for actions in their models. Nevertheless, those action semantics have been defined in the context of software development, and then do not integrate concerns from other domains like system or real time domain. Typically, actions defined by the UML specification are related to other UML concepts and since new concepts have been introduced in MARTE, we think it would have been interesting to add meaningful actions related to them and also provide associated concrete textual syntax.

V. CONCLUSION AND FUTURE WORKS

In this paper, we have discussed a co-design process called MoPCoM based on the use of the MARTE profile. For each phase of this process, we have presented the selected stereotypes of the profile and constraints related to their use. We have outlined the main encountered problems from conceptual and implementation perspective. Future works are related to VHDL and SystemC code generation after the allocation of the application on the Detailed Platform Architecture. This development is done using MDWorkbench transformation tooling from Sodius\(^2\). Further works will be carried out on modelling and implementing dynamically reconfigurable applications.

ACKNOWLEDGMENTS

The research program MOPCOM\(^3\) SoC/SoPC is supported by the French National Research Agency (ANR), the "cluster of clusters" and "Media and Networks" from Brittany and Pays de la Loire regions. The program partners are the end-user companies Thales, Thomson, Sodius as an MDA tool provider and the academic research institutions ENSIETA, LESTER, Supelec and INRIA.

REFERENCES

[CSL+] Rong Chen, Marco Sgroi, Luciano Lavagno, Grant Martin, Alberto Sangiovanni-Vincentelli, and Jan Rabaey. Uml and platform-based design.

\(^2\)http://www.sodius.com & www.mdworkbench.com
\(^3\)http://www.mopcom.fr