



VHDL-AMS Electromagnetic Automatic Modeling for System Simulation and Design

Abir Rezgui, Laurent Gerbaud, Benoît Delinchant

► To cite this version:

Abir Rezgui, Laurent Gerbaud, Benoît Delinchant. VHDL-AMS Electromagnetic Automatic Modeling for System Simulation and Design. IEEE Transactions on Magnetics, 2014, 50 (2), 10.1109/TMAG.2013.2281495 . hal-02277370

HAL Id: hal-02277370

<https://hal.science/hal-02277370>

Submitted on 21 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

VHDL-AMS electromagnetic automatic modeling for system simulation and design

A. Rezgui, L. Gerbaud, B. Delinchant

Grenoble Electrical Engineering laboratory, BP 46, 38402 Saint-Martin d'Hères, France

To improve interoperability between system level's modeling and simulation, this paper proposes an approach to export models from dedicated magnetic tools into a standardized format, such VHDL-AMS, as powerful modeling language. The goal is to let designers to use a unique modeling approach and single simulation tool to simulate the behavior of a complete electromagnetic system and to make easy the translation of existing models to VHDL-AMS. Thus preserving investment has been provided for them. The paper addresses this methodology as a computer-aided generation of the VHDL-AMS code from macro and micro electromagnetic devices for system simulation. The translation uses Model Driven Engineering (MDE) methods as the transformation of model to another and the code generation from models. Implementation and methodology are illustrated on a dynamic E-shaped actuator such a macro-system case study and a diamagnetic levitation device as micro-system one.

Index Terms — computer-aided modeling, model transformation, electromagnetic devices, system level simulation, VHDL-AMS.

I. INTRODUCTION

For the design of dynamic electromagnetic systems, several tools are developed, characterized by different capabilities of modeling for different requirements. However, an important effort is needed by system designers to integrate magnetic devices into other systems or to reuse them. This complexity makes difficult its validation regarding requirements. So, it is essential to set up a process for simulation and verification during the design of the system.

Therefore, it is important to improve the interoperability of magnetic simulation tools to make easier model exchange (Fig.1). A first approach, dedicated to component level simulation and using numerical PDE (Partial Differential Equations) solvers, is based on the weak coupling of complex or binary models (black box) [1]. For the system level simulation, a preferred solution provides an explicit model (white box) which can be strongly coupled with other white box models (mechanical, thermal, economic...). The classical way to reach such a modeling is to make a regression model based on fine simulation results [2]. However, this approach is not suit for system design since it matches only few parameters and small ranges of values.

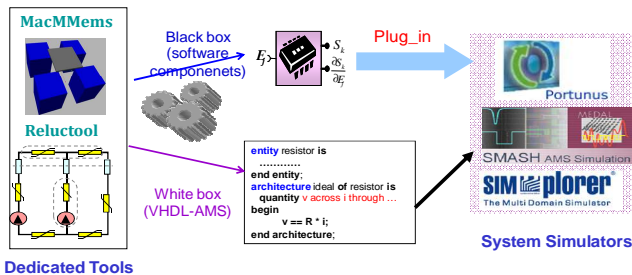


Fig.1. Process & Model interchange using VHDL-AMS

Therefore, efficient approximations based on analytical formalism are more suitable; such reluctance circuits for magnetic actuators or symbolic computation of integrals for magnetic MEMS. It would be interesting to map these models (often written in a proprietary language) to simulation languages in order to make easy their integration in simulation environments. In such context, VHDL-AMS is one of the most used hardware description language, which provides several simulation environments [3]. Indeed, dedicated tools can greatly assist the implementation of models.

In view of that, we propose an approach based on Model Driven Engineering (MDE) to translate models provided from dedicated magnetic tool to VHDL-AMS codes. We focus on the adequacy of language to describe different devices, and solutions proposed to overcome the limitations encountered.

II. PROPOSED TRANSLATION APPROACH

A. Model-Driven Engineering methods

The MDE aims to keep the models as reference point throughout the process of software development [4]. The MDE provides two main approaches for transforming models: (1) The transformation of model to another one; (2) The source code generation from models. Underlying these model representations and transformations is the notion of "meta-model", in which logical aspects are separated from physical implementation. consequently, the transformation takes an input model conform to the source meta-model and produces an output, which is conform to the target meta-model.

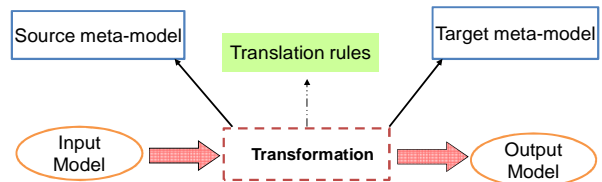


Fig. 2: Generic model transformation framework

A model transformation manipulates concepts that are specified in the source and target meta-models (fig.2). In this work, the translations, to carry out (from professional magnetic model into VHDL-AMS), depend mainly on the content of the input model : VHDL-AMS is the target model, and the sources model use very different languages and formalisms (reluctance network and Microsystems). To deal with these models, shared meta-models have to be defined for every modeling source, and a meta-model has to be defined for the target VHDL-AMS model. These meta-models describe the static structure of models manipulated by the transformation engine.

B. Proposed model translation approach

The proposed translation approach is illustrated in Fig.3. There are four mains difficulties:

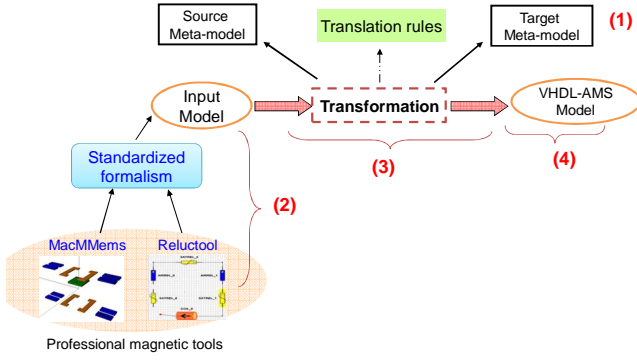


Fig. 3: The proposed approach for model translation

- (1) *Defining the VHDL-AMS meta-model* (which is the target of all our translations): The meta-model is used to check the syntactical conformity of generated models. We have developed a meta-model based on Guihal et al's work which is used as a reference due to its genericity [5].
- (2) *Defining the source model and its meta-model*: In this paper, the source models are defined in professional magnetic tools, with different modeling languages and formalisms. Indeed, SML (system modeling language used in CADES [6]) is used for the Microsystems and Java is used for the reluctance networks. We propose to use XML (eXtensible Markup Language) format, as an intermediary formalism for the transformation applied to all source models from professional tools. The source model is then specified using a standardized format based on XML-based Metadata Interchange (XMI) that supports the interchange of any kind of metadata.
- (3) *Defining the translation rules*: The goal is to transform the model inputs and produce as result a VHDL-AMS output model, conforms to its meta-model. For that, we develop a java programming algorithm to make the mapping between the two meta-models.
- (4) *Developing the VHDL-AMS code generation*: Using a *Java script*, we develop an algorithm to generate VHDL-AMS code.

This approach is implemented for modeling magnetic devices, using reluctance networks from RelucTool [7] and using integral formulation from MacMMems [8]. We will focus on the adequacy of VHDL-AMS language to describe these devices.

III. MODELING OF ELECTRO-MAGNETIC DEVICE

The reluctance network is based on the splitting up of a magnetic circuit into sub-elements (reluctances, sources...) [7]. The method, based on magnetic flux tubes, allows a simplified description of any magnetic device. So, the field distribution can be described by algebraic equations. From a reluctance description of a magnetic circuit, RelucTool generates a dedicate component for simulation and pre-sizing. It allows computing all magnetic fluxes from the element characteristics (geometry, material, sources...) [7].

We propose extensions of RelucTool by integrating VHDL-AMS generators from any reluctance network model. The implementation will be applied here on a dynamic E-shaped actuator (Fig. 4), using different manners to compute magnetic forces.

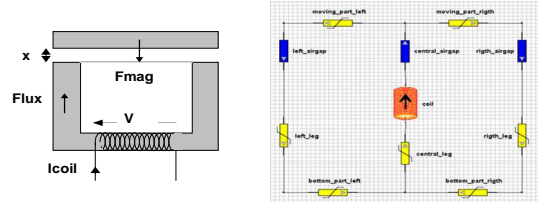


Fig. 4: Dynamic E-shaped actuator and its reluctance network model

A. VHDL-AMS translation methodology

The global methodology proposes to processes XML files, generated by RelucTool, and translates them to VHDL-AMS codes by defining transformation rules. The result is a set of instances of classes in the VHDL-AMS model. Fig. 5 presents the automatic VHDL-AMS generators developed here:

- (1) a generator for structural reluctant models that extracts components and their connections. Designers will add manually their equations for computing magnetic forces.
- (2) a generator for global behavioural models that extracts all equations from the circuit with the automatic computation of the magnetic force by the co-energy derivation.

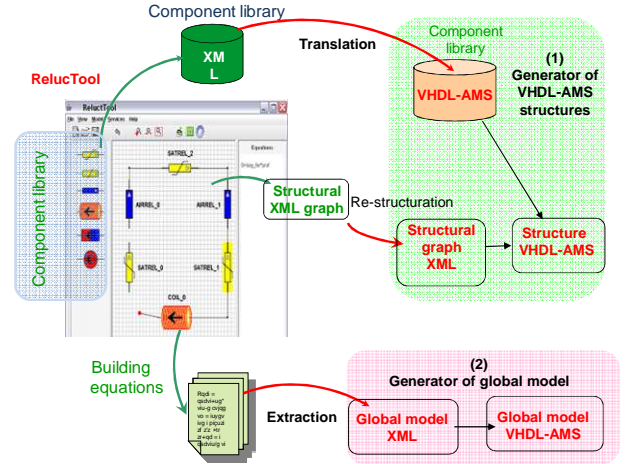


Fig. 5: VHDL-AMS translation methodology from Reluctool

B. Rules of transformation

The RelucTool XML block constitutes the basic brick to design a system. It represents a support for various elements characterizing the system: its parameters, inputs/outputs, connections, equations and its basic components. The ENTITY in VHDL-AMS represents the external view of a component with its connections. This semantic mapping allows translating a RelucTool block into VHDL-AMS ENTITY, which takes the name of the block for example.

$\{RelucTool: magnetic Model\} \rightarrow \{VHDL-AMS: ENTITY\}$

$\{Block: definition\} \rightarrow \{ENTITY: name\}$

Flow port represents the input and output I/O energetic items that may flow between a block and its environment. The same analogy is used for I/O such data and material. VHDL-AMS Port got relatively the same semantic. So, this semantic mapping allows the translation between Reluctool Flow port and I/O into VHDL-AMS Port. For example:

$\{Block: FlowPort\} \rightarrow \{ENTITY: Port\}$

$\{FlowPort: name\} \rightarrow \{Port: name\}$

$\{FlowPort: type\} \rightarrow \{Port: TERMINAL\}$

Equations and functions, defined in a block, can be translated into instructions and functions which represent the behavioral aspect of the system. Their implementation is made in the body of the ENTITY's ARCHITECTURE.

$\{Block: functions\} \rightarrow \{ARCHITECTURE: functions\}$

$\{Block: Equations\} \rightarrow \{ARCHITECTURE: instructions\}$

Components represent basic elements constituting the system. They serve to describe the composition and the internal structure of a block. Components in VHDL-AMS also represent a structural description of the circuit. So, a semantic link between parts and components is defined. The declaration of components is made in the body of the ARCHITECTURE of the VHDL-AMS ENTITY.

$\{Block: components\} \rightarrow \{VHDL-AMS: Instantiations\}$

Fig. 6 establishes the matching rules between developed (proposed) meta-models. The two translation strategies have been implemented by object-oriented programming approach.

XML	VHDL-AMS
magneticModel (nom)	Entity
parameter	generic
Inputs/outputs/Pins	ports
Components [0..N]	Architecture Instantiation
Name, type	Label entity(arch)
Parameter (name, value)	Generic map (name; value)
Inputs/outputs/Pins	Port map (name)
Links (connectors)	Port map (value) Declarative instructions (var interm)
Équations [0..N]	Declarative instructions (functions) Concurrent & Simultaneous instructions

Fig. 6: transformation rules from XML to VHDL-AMS models

A. VHDL-AMS structural reluctant models

The easy way to generate VHDL-AMS model is to translate the structure of reluctance network model, including components and connections. For that, a VHDL-AMS library of magnetic components (linear reluctances, air-gap, and magnetic coil) has been developed using XSLT based rules. The processing engine in XSLT (eXtensible Stylesheet Language Transformations) goes automatically through the document tree and applies templates as it encounters nodes. An example of rules is given in Fig. 7.

```
<xsl:template match="root">

  <xsl:for-each select="MagneticComponent">

    ENTITY <xsl:value-of select="@name" /> IS
      generic ( <xsl:for-each select="model">
        <xsl:apply-templates select="param"/>
        <xsl:apply-templates select="serial/source/param"/>
      </xsl:for-each>
      );
    PORT ( <xsl:for-each select="model">
      <xsl:apply-templates select="output"/>
      <xsl:apply-templates select="reluctance"/>
      <xsl:apply-templates select="source"/>
    </xsl:for-each>
    );
  END entity;

</xsl:template>
```

Fig. 7: Example of transformation rules with XSLT

Then, the description of reluctance networks is translated into VHDL-AMS, which means that the structure of network, with all its components (reluctances, sources) and connections is translated. Fig. 8 illustrates this mechanism for the actuator.

The automatic VHDL-AMS generator, developed in JAVA programming language, built the corresponding topology in VHDL-AMS by analyzing reluctance network provided by the user on the RelucTool GUI. This topology consists on connections of instances of components from the VHDL-AMS

magnetic library (Fig.8).

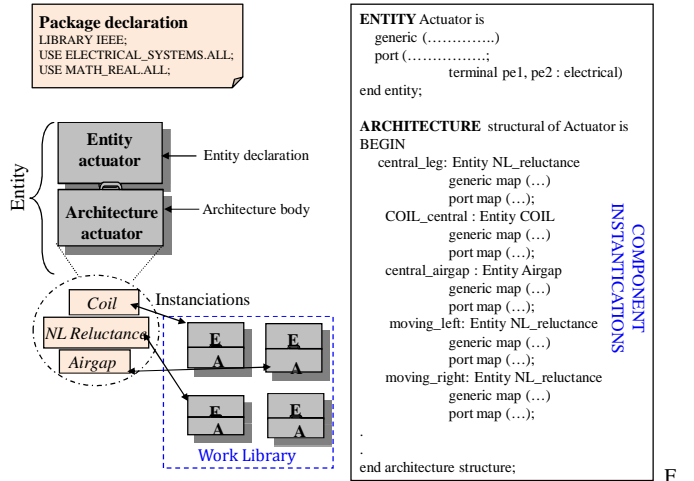


fig. 8: Structure of VHDL-AMS code of E-shape actuator

Or, The magnetic force equation can be described using virtual works. However, VHDL-AMS supports only time differentiation. A solution can be found for specific actuators using predefined energy derivation on air gap elements, but it is not generic. For that, users have to add the equations of forces into their own VHDL-AMS models.

B. VHDL-AMS global behavioral models

Another way is to build the global analytical model of the network, associated with an automated symbolic derivation which offers the computation of energy of the whole system and magnetic forces. Otherwise, RelucTool is able to generate the analytical model with forces computation. So, from the designer description of the network, RelucTool converts its elements into explicit equations and analyses the circuit topology to establish an implicit equation system. Then, all the symbolic derivatives of fluxes according to the displacement are made to complete the model with forces.

We propose a new extension of RelucTool by introducing a VHDL-AMS generator based on the export of the global analytical model with a computation of forces. The generated VHDL-AMS model applied to an E-shaped actuator has more than 200 Algebraic equations (Fig. 9).

```
ENTITY Actuator is
  generic ( ..... )
  port ( ..... ; terminal pe1, pe2 : electrical ;
        Quantity Force : out real )
end entity ;

ARCHITECTURE Behav of Actuator is

  list of intermediate variables
  list of functions
  list of derivative functions

BEGIN

  -- Equation system
  List of explicit equations
  List of implicit equations

  -- Derivatives of Equation system
  List of Derivatives of explicit equations
  List of Derivatives of implicit equations

  -- Force Equation:
  Fmag == d.coenergy_posvar ;

end architecture Behav ;
```

Fig 9: VHDL-AMS code of the behavioral model of E-shape actuator

IV. MODELING OF ELECTROMAGNETIC MEMS

The interaction between elementary models in the micro domain differs from the macro world due to the lack of ferromagnetic materials conductors of the field. The functional

gathering of Magnetic MEMS requires integral computations (Biot-Savart law, Colombian equivalent approach, magnetic-moment method) to obtain fields and forces. Complex interactions between elements have also to be formulated (target/sources, space positions and orientation of elements). This kind of interaction is not easy to be handled using the composition of VHDL-AMS components.

Therefore, we propose to use MacMMems [8], which is a dedicated tool for such system modeling. It uses elementary models (magnets and conductors) and generates parameterized semi-analytical models. We propose an extension of the tool to generate VHDL-AMS models. However, integrals formulations, which are not supported by the language, have to be used [8].

Fig. 10 shows an example of diamagnetic levitation device used to present the approach.

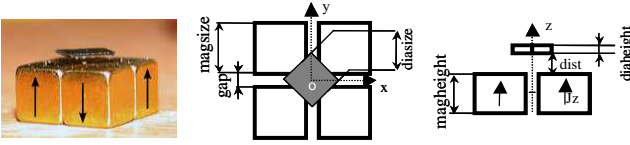


Fig. 10: Geometrical parameters of magnets and diamagnetic ball.

A. Several ways to describe a VHDL-AMS model

The global analytical modeling of the levitation supposes a negligible induce magnetic field in the diamagnetic component (very low relative permittivity). So, an analytical modeling is made in several steps:

- Compute magnetic flux from the magnets, in their own reference
- Change reference to the levitation element
- Compute the global magnetic field received by the target, using the superposition theorem (sum of fields from magnets)
- Compute the force applied to the levitation element.

$$\vec{F} = \frac{\mu_0}{2} \cdot \iint_s [\vec{\chi}]^n \cdot |\vec{H}_{ext}|^2 ds$$

MacMMems implements such a modeling, in a generic way. A VHDL-AMS description of such a modeling needs simultaneous declarations or functions. VHDL-AMS allows only time integration. For spatial integration, the use of numerical function is needed as shown in Fig. 10. So, the main problems to deals with MEMS modeling are:

- dynamic calls of a function
- numerical spatial integrations

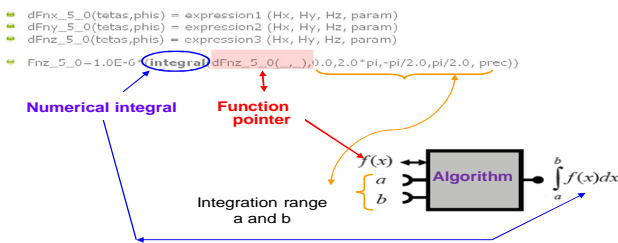


Fig. 11: numerical integration of function in SML language (CADES)

VHDL-AMS aims to define and design physical systems only. Computation concepts are separated from the language, so a pointer to a function is not available for a function (only pointers on data are possible). A solution consists in using a

dynamic compilation, i.e. the code of the integrals is compiled at every time it is used. Here, the integral can be defined directly in VHDL-AMS, using successive spatial derivations [10]. Such an approach may increase strongly the model size according to the discretizing.

To improve such a modeling, several solutions are provided

- Use of PROCEDURAL simultaneous instruction: [10]
- Use of GENERATE instruction [10]
- Implement specifically integral method for each function, so duplication of the integral algorithm
- Use of FOREIGN instruction [11]. This last instruction allows to deals with external functions. Its implementation is specific to the VHDL-AMS compiler, and few compilers implement it.

In this paper, the two last solutions have been implemented. For the FOREIGN approach, Smash has been used; external functions are connected using C programming such in [12].

V. CONCLUSIONS

The paper proposes an automatic model generation methodology to create VHDL-AMS models from two dedicated tools for magnetic systems: RelucTool (reluctance networks) and MacMMems (magnetic MEMS). In this work, different solutions to export models to VHDL-AMS are presented. An automatic generator from RelucTool have been developed and applied to a dynamic E-shaped actuator. A methodology of exporting behavioural of magnetic MEMS from MacMMems is described and presented on a diamagnetic levitation. The whole methodology is based on MDE (Model Driven Engineering) methods.

In future work, those automatic generators should be performed and extended to deal with more complex modeling systems.

REFERENCES

- [1] A. Rezgui, et al. "Unified modelling technique using VHDL-AMS and Software components", ELECTRIMACS 2011, 6-8th June 2011, Cergy-Pontoise, France
- [2] P. Schwarz, P. Schneider "Model Library and Tool Support for MEMS Simulation", Conference on "Microelectronic and MEMS Technology" Edinburgh 2001.
- [3] VHDL-AMS: VHDL Analog and Mixed-Signal Extensions, IEEE Standards 1076.1-1999, Mar. 1999.
- [4] J. Bézin, "Introduction to Model Engineering", 2006 ATLAS Group (INRIA et LINA) Nantes
- [5] D. Guibal, et al. "Mixed Design of Integrated Circuits and System", Conference on Mixed Design of Integrated Circuits and System, Gdynia, 2006, pp 549 – 554.
- [6] P. Kauffmann, et al. "Efficient multipoles modeling for linear magnetized beads manipulations" manuscript, "Scientific and Clinical Applications of Magnetic Carriers, Rostock : Germany, 2009
- [7] B. du Peloux, et al. "Automatic generation of sizing static models based on reluctance networks for the optimization of electromagnetic devices," IEEE Transaction on. Magnetic, 2006.
- [8] H. L. Rakotoarison, et al. "Methodology and tool for generating semi-analytical models Used to pre-design electromagnetic MEMS", Conference on "Electromagnetic Field Computation", 2006
- [9] Guelaz, R, et al. "Ultrasonic piezoceramic transducer modeling with VHDL-AMS IEEE 1076.1", Proceedings of IEEE Sensors, 2004, 24-27 Oct. 2004, pp 87 - 90 vol.1
- [10] P. J. Ashenden, et al., "The System Designer's Guide to VHDL-AMS : Analog, Mixed-Signal, and Mixed-Technology Modeling", Morgan Kaufmann Publishers, San Francisco, 2002
- [11] SMASH, "Foreign C functions in VHDL-AMS and SPICE models", Release 5.13, 2009

- [12] A. Rezgui, et al. "VHDL-AMS to Support DAE-PDE Coupling and Multilevel modeling", IEEE transactions on magnetics, February 2012, vol. 48, Issue. 2.