Development of a Bond Graph Based Model of Computation for SystemC-AMS

Torsten Maehne, Alain Vachoux, and Yusuf Leblebici
Laboratoire de Systèmes Microélectroniques (LSM), École Polytechnique Fédérale de Lausanne (EPFL)
EPFL/STI/IEL/LSM, Bâtiment ELD, Station 11, CH-1015 Lausanne, Switzerland
Phone: +41(21)69-36922, Fax: +41(21)69-36959, WWW: http://lsm.epfl.ch/, E-Mail: torsten.maehne@epfl.ch

***Abstract*—The modelling and simulation capabilities of SystemC-AMS concerning conservative continuous time systems involving the interaction of several physical domains and with digital control components are currently limited. Bond graphs unify the description of multi-domain systems by modelling the energy flow between the electrical and non-electrical components. They integrate well with block diagrams describing the signal processing part of a system. The goal of this work is to integrate the bond graph formalism as a new Model of Computation (MoC) into the SystemC-AMS prototype.**

I. Introduction

The advances in processing technologies supports since several years the trend towards more and more feature rich and heterogeneous Systems-on-Chips (SoCs) or Systems-in-Packages (SiPs). The increasing complexity of the implemented functionalities, the diversity of possible implementations (e.g., digital/analog/RF hardware, software, Micro-Electro-Mechanical System (MEMS)), and the rapid evolution of the needs (e.g., time to market, product diversity, compliance to standards) ask, on the one hand, for early partitioning decisions that meet design constraints and, on the other hand, for easy reuse and retargeting. To cope with the resulting design complexity, application domain dependent modelling and abstractions concepts need to be used in parallel throughout the design flow.

SystemC [1] is a C++ library, which allows to model complex digital hardware/software systems by mapping them on communicating processes, which are executed and synchronized by a *Discrete-Event* (DE) MoC based simulation kernel. There have been several attempts to extend SystemC to support the design of heterogeneous systems as described earlier. *SystemC-A* [2] and *SystemC-AMS* [3] use a SPICE-like approach to model energy conserving systems using generalized networks, which limits possible improvements of the simulation performance. *SystemC-AMS* also implements the *Synchronous Data Flow* (SDF) MoC to model signal processing dominated continuous time behavior. A synchronization layer handles the communication between both continuous time MoCs and the discrete event MoC of the SystemC simulation kernel. This is an advantage over *SystemC-A*, which requires modifications to the standard SystemC kernel thus hampering the integration of other extensions. *SystemC-WMS* [4] uses another approach based on the Wave Digital Filter (WDF) theory by describing the conservative components through a scattering matrix and their interaction through incident and reflected energy waves. The implementation is entirely based on the hierarchical channels of SystemC. This implies the scheduling of many discrete events per analog solution point and thus limits the simulation performance.

This work has been funded by the Hasler Stiftung under project № 2161.

This work aims for improving the modelling and simulation capabilities of SystemC-AMS regarding conservative continuous time components from different physical domains and their interaction with discrete time (digital) control components by implementing a new MoC based on the *bond graph formalism*. Section II introduces this formalism and Section III describes the resulting requirements and first results of its integration as a new MoC into SystemC-AMS.

II. Bond Graph formalism

The bond graph formalism [5] unifies the description of multi-domain systems and is mostly used in mechanical engineering, mechatronics, and control theory. Each domain-specific system model (e.g., electrical circuit, mechanical multi-body system, fluidic or thermal networks) can be transformed into a bond graph representing the energy flow between generalized elements of a multi-domain system. The energy link between the ports of two elements El_i and El_k is represented with an half-arrow shaped *bond*:

$$El_i \xrightarrow[f]{e} El_k$$

Associated to each bond are an *effort* e and a *flow* f *variable*. They are called *power variables* because their product is the power P. By definition, the half-arrow points into the direction, in which the power flows for positive e and f. The description of dynamic systems requires also to introduce two *energy variables*, called *(generalized) momentum* $p(t)$ and *(generalized) displacement* $q(t)$, which are defined as the time integral of an effort and flow, respectively:

$$p(t) = \int^t e(t)\,dt \qquad q(t) = \int^t f(t)\,dt \tag{1}$$

The energy E can be expressed as a function of time or of one of the energy variables:

$$E(t) = \int^t P(t)\,dt \quad E(q) = \int^q e(q)\,dq \quad E(p) = \int^p f(p)\,dp \tag{2}$$

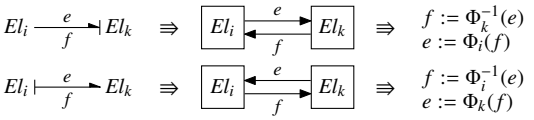
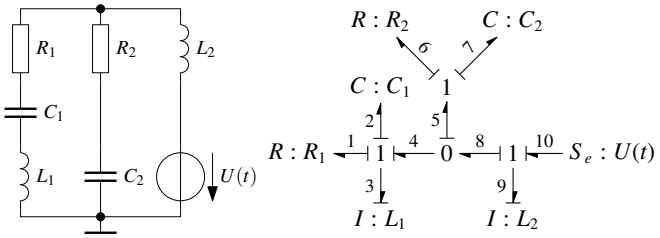


Fig. 1. Interpretation of a bond as a bilateral signal flow

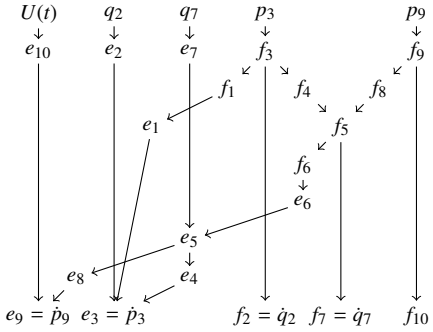
The definition of the *generalized* power and energy variables is independent of a particular physical domain. The link to the physical domain is kept through the units attached to the variables and parameters of the generalized components. Effort and flow can, e.g., represent voltage/current, force/velocity, or pressure/volume flow rate in electrical, mechanical, or hydraulic systems, respectively.

The energy exchange through the bond between two elements causes the effort and flow variables to act in opposite directions. This can be used to determine the computational direction, which is indicated by a perpendicular stroke at one end of the bond. This *causal stroke* states that at this side the effort variable e is known (it acts as an input) and f can be calculated as a function $f := \Phi_k^{-1}(e)$. Consequently, the flow f is known on the other side of the bond and acts as an input to a function to calculate the effort: $e := \Phi_i(f)$ (Fig. 1). The equations describing the component behavior impose a *required*, *preferred* (e.g., due to numerical reasons), or *free* causality (*effort-in* or *effort-out*) on the element ports. The resulting constraints need to be propagated to all related ports. Methods like the Sequential Causality Assignment Procedure (SCAP) [5] exist to systematically define the causality of a bond graph. This and the compact visualization of the energy flow and computational structure are their main advantages. The assigned causalities allow to sort the element equations in the right order for an efficient model execution and to do some further formal checks on the model: the number of states and non-states in the system, the presence of algebraic loops during model execution, or if it is an ill-posed model.

Three generalized 1-port elements represent the resistive R, inertial I, and capacitive C elements independent of the considered physical domain. Energy sources are modelled as effort source S_e and flow source S_f elements. Quantity transformations (also across domain boundaries) are represented through the transformer *TF* and gyrator *GY* 2-port elements, which ensure the conservation of energy. Multi-port junction elements represent explicitly the connexion of elements, which are exposed to a common effort (0-junction) or a common flow (1-junction). All elements may have non-linear characteristic equations. *TF*, *GY*, S_e, and S_f can be modulated by an external signal. As an example, Fig. 2 shows a simple electrical circuit, its equivalent bond graph annotated with causality, and the derived computational structure. The bond graph is constructed from the circuit by using the equivalences between the electrical and bond graph elements and 0-/1-junctions to represent the parallel/serial connections of the circuit, respectively. The computational structure results from the dependencies between the quantities due to the element equations, the bond graph structure, and the assigned causality.



(a) Simple electrical circuit (b) Bond graph with annotated causality



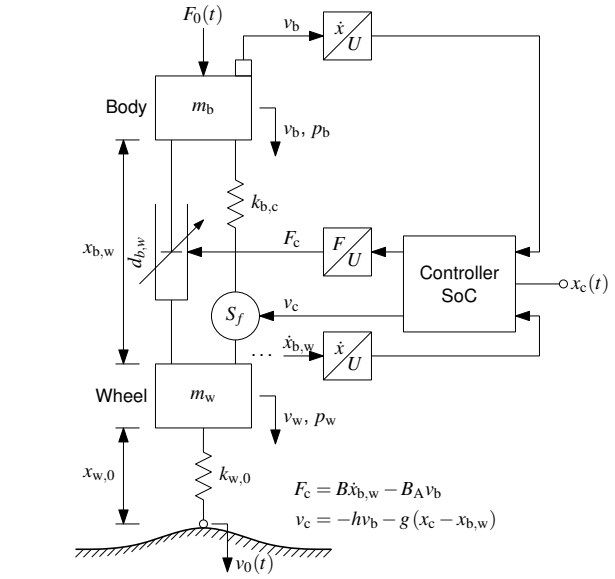
(c) Dependency graph representing the computational structure of the bond graph

Fig. 2. Example of the transformation of an electrical circuit to an equivalent bond graph annotated with causality and then derived computational structure

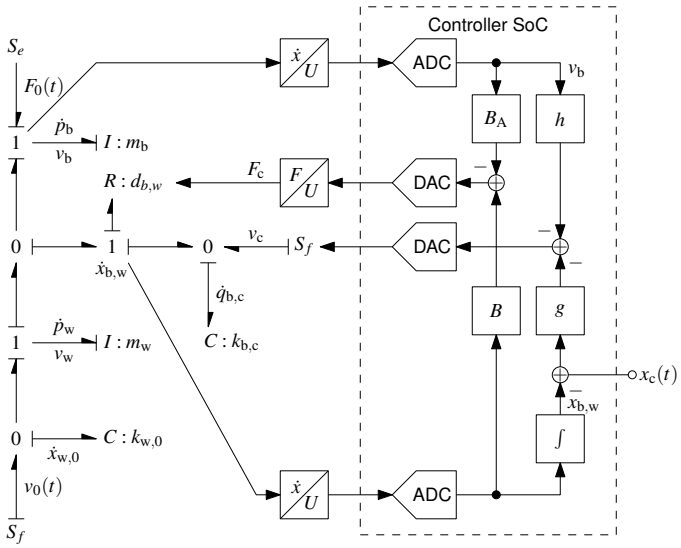
The causality assignment allows also for a natural integration of bond graphs with signal flow graphs. An example is given in Fig. 3 in the form of a car wheel model of an electronically controlled suspension system incorporating a damper and a load-leveller [5]. The system is once modelled in the "classic" domain-specific way and once using bond graphs for the energy conserving part and block diagrams for the signal processing part. The blocks in the signal flow graph can take power or energy variables as input and can modulate the sources or element parameters of the bond graph.

Complex systems such as Analog and Mixed-Signal (AMS)-SoCs require a hierarchical description of their structure. This can be achieved by using the more abstract *word bond graphs* [5], which add the usage of "macros" to the classic bond graph modelling to represent a multi-port component of the overall system. Such systems are also often characterized through the presence of digital units, which control the continuous time parts through feedback and switching of energy flows due to digital signals. The switching can be modelled using idealized controlled junctions. Their presence indicate a *hybrid bond graph* [6]. An active junction constrains the causality of attached bonds like a normal junction. When it gets inactive, it imposes a zero on the power variable, which is common to all attached bonds, leading to a causality change. In electrical terms this means that a parallel connexion modelled by a 0-junction is "short-circuited" and a series connexion modelled by a 1-junction is "left open". The causality change needs to be propagated into the graph and requires a solver reinitialization imposing a simulation performance penalty.

Specialized bond graph tools such as MTT, 20-sim, and En-



(a) "Classic" domain-specific system representation



(b) Representation coupling a bond graph for the energy conservation part with a block diagram for the signal processing part

Fig. 3. Car wheel model of an electronically controlled suspension system incorporating a semi-active damper and a fast load-leveller [5]

port (reviews available on, e.g., http://www.bondgraphs.com/) are not popular in the microelectronics community. There are also efforts to integrate bond graph support in widely used mathematical tools such as MATLAB/Simulink (e.g., BondLab) or Mathematica (e.g., Bond graph tool box), which usage is limited to very early design stages. Another approach is to use the capabilities of an AMS-HDL to represent bond graphs. BondLib [7] implements causal/a-causal bond graphs support for Modelica/Dymola. However, Modelica's capabilities are weak on the discrete event side, which is one reason why it is not suitable for AMS-SoC design. [8] shows that bond graphs can be represented in VHDL-AMS—however, not very efficiently as equations are duplicated due to their mapping to generalized networks. Those impose Kirchhoff's voltage and current laws, which are already implicitly considered through the 0- and 1-junctions. Causality cannot be assigned to the bonds and thus not be benefited from. All three aspects have a negative impact on the simulation performance of the model.

III. SystemC-AMS Bond Graph Extension

The previous section showed that the bond graph formalism has some properties, which make its application to AMS-SoC design attractive and would complement well the already available MoCs in SystemC-AMS. This motivates the development of a bond graph MoC. These properties are:

- The unified description of conservative multi-domain systems with dimension units attached to variables/parameters keeps the link to the related physical domain. A dimensional analysis can thus ensure proper model assembly. This is currently not possible in SystemC-AMS since all non-electrical conservative behavior needs to be mapped to electrical primitives of the Electrical Linear Network (ELN) MoC discarding the dimension units.
- The causality analysis of a bond graph allows the ordering of the describing equations for an optimized (procedural) execution (similar to the static scheduling of the Synchronous Data Flow (SDF) MoC). This alternative to the set-up and solution of a global Differential Algebraic Equation (DAE) system for a generalized network description promises a higher simulation performance.
- The semiformal checks (number of states/non-states, algebraic loops, ill-posed model) possible after causality analysis offer insight into the model's physical and computational structure to the designer.
- Word bond graphs permit hierarchical modelling to tackle complexity and allow reuse.
- Bond graphs integrate well with signal-flow models (SDF MoC) and allow switching of energy flows due to external signals (DE MoC).

To retain these advantages and implementing them in a way coherent to the philosophy and modular architecture of SystemC-AMS are the principal design requirements to the bond graph MoC [9]. The MoC is in an early development stage. This includes the definition of how the layered architecture of SystemC-AMS is going to be extended (Fig. 4) and the start of the design and implementation of the necessary class hierarchy. The *module layer* provides the user interface to the library. Templated `sca_bg_port` and `sca_bg_bond` classes will allow to interconnect instances derived from `sca_bg_module`, which is the base class to implement the bond graph primitives. The template nature of the port and bond classes allows them to be typed to not only carry **`double`** quantities but properly typed quantities with a dimension unit. It is planned to use the *quantitative units library*, which is part of the *Boost project* (http://www.boost.org/) and implements units not only as "syntactical sugar" but to use them for compile time dimensional analysis. Fig. 5 shows how this translates into a proposed syntax for the structural description of a simple bond graph with an `sc_module`. The *view*

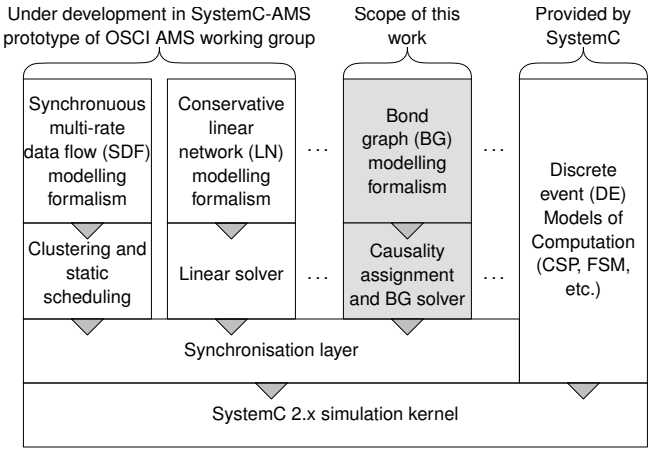
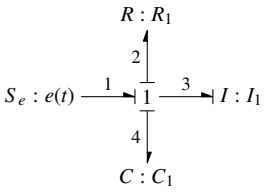


Fig. 4. Architecture of SystemC-AMS showing the MoCs provided by the SystemC kernel and the AMS extension as well as the synchronization layer

layer implemented by `sca_bg_view` will cluster the interconnected `sca_bg_module`s. The therefore necessary flattening of the hierarchy is handled by the SystemC kernel. The causality analysis, formal checks, and ordering of the equations of the bond graph primitives is then done per cluster. The *solver layer* executes the ordered equations, which will be handled by an `sca_bg_solver` object per cluster. The scheduling and *synchronization layer* executes the `sca_bg_solver`s as part of the surrounding SDF cluster. It also notifies the solver object of switching events causing an energy flow commutation. This will toggle causality reassignment and regeneration of the computation model at runtime. Their efficient implementation is still topic of active research [6].

IV. Conclusions and Outlook

The reviewed current state of AMS extensions to SystemC shows a gap between low-level SPICE-like modelling of conservative behavior and high-level signal-flow modelling of non-conservative behavior. The presented work addresses this issue through the development of a bond graph MoC for SystemC-AMS to improve its modelling and simulation capabilities in the field of energy conserving multi-domain components and their interaction with digital control components. The design and implementation of the bond graph MoC for SystemC-AMS are under way. A first definition of its architecture and a preliminary syntax for describing bond graphs were presented. In a first step, the "classic" bond graph primitives will be implemented. Future working directions are an optimized implementation of hybrid bond graphs controlled by discrete events, the optimization of the computational structure to reduce the number of algebraic loops, and further verification of the models using the results from causality analysis.



(a) Bond graph of `simple_bg`

```
#include "sca_bond_graph"

using namespace sca_bg;
using namespace sca_bg::domains;
using namespace boost::units;
using namespace boost::units::SI;

struct simple_bg : public sc_module {
  // bond graph elements
  sca_R<electrical> R1;
  sca_C<electrical> C1;
  sca_I<electrical> I1;
  // sine source derived from sca_Se
  Se_sine<electrical> Se1;
  // junctions and bonds
  sca_1<electrical> CFJ1;
  sca_bond<electrical> b1, b2, b3, b4;

  // set element parameters
  simple_bg(const sc_module& name)
  : Se1("Se1", 0.0 * volt, 1.5 * volt,
        50.0 * hertz, 0.0 * radian),
    R1("R1", 1.0e3 * ohm),
    C1("C1", 10.0e-9 * coulomb / volt),
    I1("I1", 1.0 * micro * henry),
    CFJ1("CFJ1"),
    b1("b1"), b2("b2"), b3("b3"), b4("b4")
  {
    // connectivity
    b1.bind(Se1.p, CFJ1.p);
    b2.bind(CFJ1.p, R1.p);
    b3.bind(CFJ1.p, C1.p);
    b4.bind(CFJ1.p, I1.p);
  }
};
```

(b) Proposed syntax

Fig. 5. Structural description of a simple bond graph with an `sc_module`

References

[1] *IEEE Standard 1666-2005, SystemC Language Reference Manual.*

[2] H. Al-Junaid, T. Kazmierski, and L. Wang, "SystemC-A modeling of an automotive seating vibration isolation system," in *Forum on Specification and Design Languages (FDL) 2006.*

[3] A. Vachoux, C. Grimm, and K. Einwich, "Extending SystemC to support mixed discrete-continuous system modeling and simulation," in *IEEE International Symposium on Circuits and Systems (ISCAS) 2005.*

[4] S. Orcioni, G. Biagetti, and M. Conti, "SystemC-WMS: A wave mixed-signal simulator," in *8th International Forum on Specification & Design Languages (FDL) 2005.*

[5] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems*, 4th ed. Wiley, 2006.

[6] C. D. Beers, E.-J. Manders, G. Biswas, and P. J. Mosterman, "Building efficient simulations from hybrid bond graph models," in *2nd IFAC Conference on Analysis and Design of Hybrid Systems 2006.*

[7] F. E. Cellier and R. T. McBride, "Object-oriented modeling of complex physical systems using the Dymola bond-graph library," in *6th SCS International Conference on Bond Graph Modeling and Simulation (ICBGM) 2003.*

[8] F. Pêcheux, B. Allard, C. Lallement, A. Vachoux, and H. Morel, "Modeling and simulation of multi-discipline systems using bond graphs and VHDL-AMS," in *International Conference on Bond Graph Modeling and Simulation (ICBGM) 2005.*

[9] T. Maehne and A. Vachoux, "Proposal for a bond graph based model of computation in SystemC-AMS," in *Forum on specification & Design Languages (FDL) 2007.*