VHDL-AMS: The Missing Link in System Design - Experiments with Unified Modelling in Automotive Engineering

Eduard Moser, Norbert Mittwollen
Robert Bosch GmbH, FV/FLI
Postfach 106050
D-70049 Stuttgart, Germany
Email: moser@fli.sh.bosch.de

Abstract

After the IEEE ballot accepted the first draft language reference manual for VHDL-AMS (IEEE PAR 1076.1) in October 1997, we now can spend time and effort on applying the new arising methodology to real world problems outside the electronic domain. In automotive engineering we have system design problems dealing with hydraulic or mechanic components and their controlling units, for which we expect a major advantage by introducing unified modelling to all domains.

With the Brite/EuRam-Project TOOLSYS *(a joined effort of automotive industry and tool makers to apply VHDL-AMS as unified modelling language on mixed-domain applications) we prove the suitability as unified modelling and interchange language for real-world systems and components.*

First experiments with hydraulic components reveal numerical problems on analog circuit simulators. None of the available strategies for these particularly hard problems are included by the electronic simulator makers. With VHDL-AMS multi-domain modelling seems possible, now we need multi-domain simulation environments.

1 Introduction

Languages to describe simulation objects like physical components or control algorithms are usually based on differential algebraic equations. However, the proprietary languages available are all uncompatible, and due to tool dependent features reflected in the languages no method exists for automatic conversion. Since all model libraries have to be rewritten and the designers need basic training, the introduction of a new tool is extremely expensive. The exchange of models within the company as well as with other companies usually needs lots of extra work. A unified modelling language would solve most of the problems concerning behavioural description. The advantages of such a language have already been discussed in detail [2].However, multi-dimensional geometrical models like multi-body descriptions are not covered by this approach.

Due to its mathematical foundation, VHDL-AMS (VHDL Analog and Mixed-Signal) is suited to express plant models and control algorithms needed for car makers and suppliers. It is able to communicate maker's requests and supplier's solutions. Furthermore, the language is suited for model exchange within the companies. The language will have tool support and industrial relevance soon, since it already is the language of choice for the electronic domain.

The MSR consortium, a group of German car makers and suppliers, has been founded in order to find some unified methods for the easy exchange of product specifications [8]. One task was to find a unified language for behavioural modelling of controllers and plants.

The MSR decision to use VHDL-AMS as common model interchange format is based on a thorough study [3] including many typical examples written as VHDL-AMS code. Besides the language for the easy exchange of product specifications concerning the behaviour of controller and components we hope for a unified simulation environment capable of solving the complex mixed domain descriptions.

The *TOOLSYS*-Project as a joint effort of car industry and software supplier verifies the unified approach to relevant applications. We want to prove the idea not only for component design within particular domains (hydraulics, mechanics) but also for system and control design. For the control design we do not expect one simulator to take over the entire description of all components and simulate the system as a whole containing every detail included in the component models. In order to understand everything sufficiently it is, in our opinion, necessary to perform some abstraction and to derive abstract system models for every detailed component model.

Throughout this paper we relate to VHDL-AMS as defined in the draft language reference manual [14] and the introductions given in [1, 13].

The next section explains the unified modelling. It characterises the area of interest for the automotive industry and includes a problem description as well as a discussion on other approaches discussed so far. The section 3 gives a brief overview on the methodology which is designed for the *TOOLSYS* project. The paper closes with sections on mixed domain simulation and describes the problems with hydraulic simulation performed on electronic simulators.

2 Unified Modelling Language

In 1987, VHDL became a standard and a unified modelling language for digital hardware. For system design and control engineering the "Unified Modelling Language" must be capable of covering a much wider area of disciplines including physical components (e.g. analog circuits, hydraulics), control algorithms and digital components.

2.1 What is the problem

All the different languages used within the area discussed above almost contain similar concepts. All of them show a way to express differential algebraic equations (DAE). But what does the simulators do with them? What divergence from the "true" result will be accepted as solution (concept of tolerance)? Only few languages have explicit concepts for discontinuities, but even less is known how they perform. What will be done in case of exceptions (e.g. non convergence)?

Opposite to digital systems with only discontinuous value changes at discrete time points the numerical algorithm solving the continuous description only allows to approximate to some degree. Any unified language, which may allow further development on products and solution strategy, cannot prescribe the numerical algorithm and, therefore, has to be defined using abstract mathematical notions for the ideal solution and permitted divergences.

For a better understanding of modelling languages we analysed a set of models written in various proprietary languages. All these languages have little tricks for missing concepts. One such example describes the "threshold-crossing" on a simple control engineering simulator:

```
if ( x * xold < 0.0 ) then ...;
xold := x;
```

The trick assumes a bounded step size and performs quite well with explicit integration methods. It is impossible to detect the construct "threshold-crossing" from the example. Further examples relied even more on knowledge of the particular simulation engine.

2.2 Approaches towards unified modelling languages

There have been many approaches on unifying modelling languages for the domains so far. But besides VHDL-AMS all the approaches only touch certain subdomains like control engineering or digital electronics or recently digital and analog electronics. Only some approaches shall be mentioned as follows.

2.2.1 Control engineering approaches

For system simulation more than 30 years ago a unified modelling language has already been developed by the Simulation Software Committee of Simulation Councils, Inc. (SCi) called Continuous System Simulation Language (CSSL) [12]. One such language which follows these recommendations is ACSL [6].

CSSL focuses only on system simulation and does not address the needs of physical systems (particularly electronics with conservative communication and highly structured components).

Another approach to control engineering did not try to unify the modelling itself, but provided a unified interface language called *DSblock* [9, 5]. Converters between many proprietary languages and the interface language are available.

2.2.2 Unified modelling language C - no solution

Since C has been used in many simulators as modelling language, it is proposed to use the programming language for unified modelling. But it does not solve any of the problems above. C is valuable only for coding expressions and functions (e.g., for explicitly coding the right-hand side of ordinary differential equations, but the same code could be the transition function of a state machine). It does not have semantic concepts for coding DAE, discontinuous behaviour, threshold detection, structural decomposition with conservative and non conservative communication, etc. Therefore, if C is used, it has to be extended with these and other concepts.

2.2.3 VHDL-AMS

VHDL-AMS is the only approach which unifies digital and analog electronics with system design aspects for the first time. It is based on VHDL and, therefore, has already a precise definition of all issues as far as a regular programming language is concerned (e.g., expressions, functions). Furthermore, it provides means for structuring systems and communication. The power for describing complex discrete time systems might be the only drawback, since a VHDL-AMS compliant simulator has to support full VHDL. Special purpose simulators (e.g., for hydraulics or real time) may only support some language subset including simple synchronous processes.

The extension AMS covers all the missing concepts for a full description of components and systems according to the definition in section 2. The original design as a unification of digital hardware description (VHDL) and analog circuit description and its additions for describing system behaviour by adding features for handling discontinuities (*BREAK* statement, which allows to re-initialise individual quantities and to restart of the analog kernel, detecting of

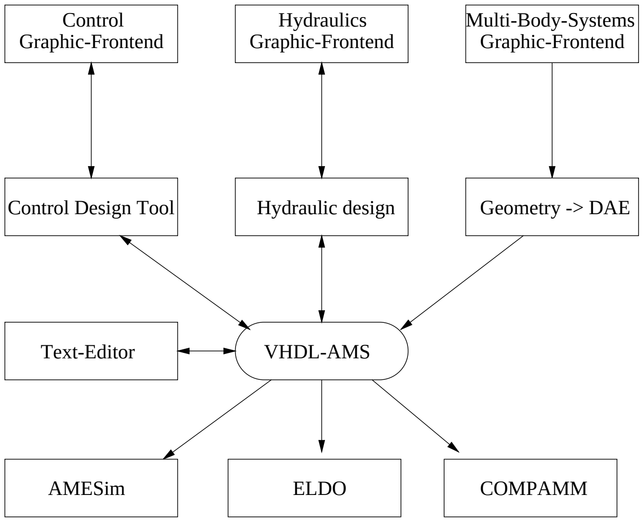

Figure 1: TOOLSYS model exchange using VHDL-AMS

threshold crossing) as well as non-conservative communication links, are now well suited for the entire domain.

3 TOOLSYS design methodology

Within the Brite/EuRam-Project *TOOLSYS*[1] partners from automotive industry (PSA, VOLVO, Bosch), tool makers (ANACAD, IMAGINE, CEIT) and research (CNRS/LAAS) will prove the capability of VHDL-AMS as multi-domain representation language on real-world applications (Figure 1).

The applications like a brake system with ABS are modelled including their aspects from hydraulics, mechanics and control. Within the first stage, the specialists will design the particular model aspects on the domain specific simulators, later these models shall be exchanged and combined on all simulators.

Besides the development of complicated and detailed models, featuring special effects necessary for the component designer, each partner is obliged to design simple models featuring only the basic behaviour depending on very few parameters. These simple models should be sufficient for the design of other components linked to them. Simple models might be the result of a system analysis of the components and can be used as reference for further model development. They might be used during the development of an appropriate test bench. If no system analysis is performed and no abstract models are available, these models have to be generated by analysing the detailed component descriptions. From our viewpoint, it seems necessary to have abstract understanding of the components for efficient control and system design. The unified modelling approach will help calibrating the abstract model parameters since both models can be executed on the same simulation engine.

The project will extend our experience using VHDL-AMS within different domains as well as the tool makers experience about the domain specific problems coming up within other than the simulators native domain. Since each of the domains has its own highly advanced simulation techniques, experts have to deal with the specific problems.

For us as users we do not expect that every simulator will solve all the specific numerical problems arising from foreign domain specific conditions. We do expect from the unified approach an easy understanding of the different domain models and an easy exchange of simplified models with only moderate numerical demands.

4 Unified Simulation

After having one such language to design any DAE based model description one might expect system simulation is easily achieved by using one simulator and plugging together all the different domain models. But for now there is no such simulator available, since all the simulators are constructed on the base of particular needs.

The aspects are how the simulator handles huge systems of equations and how the simulator behaves, if the regular strategy fails (e.g. discontinuities, non convergence). Some of the domain specific characteristics are described as follows:

Electronic circuit simulation has been optimised for coping with huge numbers of transistors (e.g., 100 000 equations). High index problems or discontinuities do arise rarely.

Hydraulic simulation does appear almost ever in combination with mechanic components (e.g., valves), the combination produces extremely stiff systems. Due to sudden state changes (e.g., liquid/gas) the system behaviour is highly discontinuous. Furthermore, the simulation has to cope with stationary low frequency modes with low damping. The described systems are rather small but extremely complex.

Multi Body System descriptions are written as a network of constraints highly depending on each other. Therefore, the simulation has to cope with high index problems.

System simulation is very difficult to characterise. Since the domain specific characterisation can be passed on to the system level, but usually it is expected that a system model of a component contains only a simple description of the general behaviour of the whole component not expressing every detail. From experience we can find discontinuities fairly often in such

[1] Contract BRPR CT96 0303

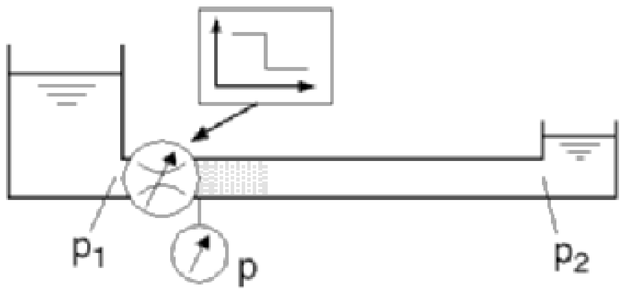

Figure 2: Experiment to demonstrate hydraulic effects, L=200 m, D=15.2 mm, fluid=water

system models (simplifications of complex physical state changes can be easily expressed using discontinuous behaviour e.g. stick/slip-friction).

Concluding, it should be said that none of the solutions for one domain can be sufficient for the other domains. Further development will provide mixed-domain simulators with true support for more then one domain.

5 Experiments with Hydraulic Systems: Pipe

Advanced simulation techniques to solve huge transistor networks but without support for the hydraulic specific problem domain can only be used for system simulation with hydraulic components included but will perform poorly on hydraulic component design.

5.1 What makes hydraulic systems so hard to simulate

Variable stiffness: According to [10], the stiffness index varies from < 1000 (non-stiff, pneumatic systems) up to 10^{12}.

Discontinuities: The models will include discontinuities due to discontinuous behaviour of mechanic devices, being part of the hydraulic description, and because of cavitation as well as laminar-turbulent flow transition. Cavitation produces high pressure peaks because of collapsing gas bubbles within pipes filled with liquid.

Low frequency stimulation like engine rotation can in cases of lowly damped resonance build up large amplitude oscillations, which may cause damage and other undesired effects depending on the amplitude. For getting reliable simulation results with precious values for the amplitude of these oscillations, long range transient simulations have to be performed.

Analog circuit simulators provide only very few implicit integration methods, usually Gear and Trapezoidal

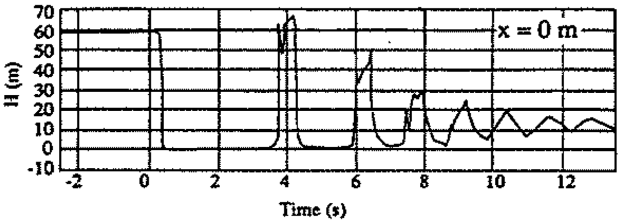

Figure 3: Experimental measurements on the pipe in Figure 2

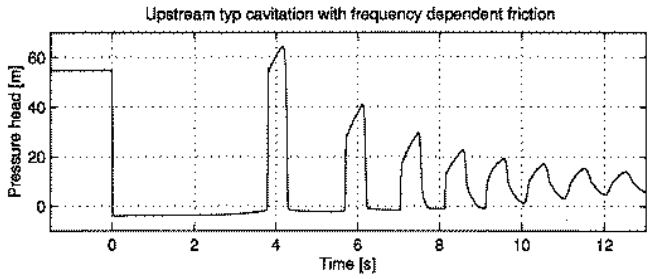

Figure 4: Simulation results on the pipe in Figure 2

with fixed order. Low order methods guarantee higher stability, but are CPU time consuming during intervals of low frequency stationary oscillation. High order methods, on the other hand, cannot be used during intervals of high frequency. Hydraulic systems with variable stiffness, with discontinuities and low frequency oscillation are efficiently simulated using adaptive integration schemes (e.g., DASSL or LSODA adapt the order and the local error [4, 10]).

In order to obtain a reliable result from an analog circuit simulator, one has to set the local errors to very small values. In many cases we could not obtain any solution at all. The long range transient simulation combined with regularly occuring discontinuities finally leads to unbearable CPU times (by a factor of 10 - 100 longer compared to a domain optimised simulator). Other methods than transient analysis were not considered, because of the discontinuities.

Experiments with cavitation in pipes (e.g., Figure 2) showed all the described effects [11] (our numerical experiments were carried out related to [7]). Two constant pressure sources with a high pressure of $p_1 = 5\ bar$ and a low pressure of $p_2 = 0.98065\ bar$ are connected through a pipe. Near the high pressure source, a valve can be instantaneously opened or closed (we are not concerned with the correct valve model). After switching the valve, a pressure wave travels from left to right. The wave pulse will be corrupted due to cavitation and, therefore, will produce discontinuities. Cavitation occurs since the wave pulse would lead to negative pressures, which are not sustained in hydraulic fluids. In other physical domains, the trajectory of the wave pulse could have negative values for some time, but potential negative pressures will cause a state change from liquid to gas. The resulting behaviour of the sequence of discontinuous state changes can only be expressed in average, e.g., Figure 4. Due to reflections on both ends of the pipe and due to viscous damping, the wave pulse will finally converge into a smooth wave. Figure 3 shows only the first stimulation and the smoothly decreasing wave.

The modelling problem of the pipe is under investigation, but no satisfying solution is known so far. One approach is based on describing the pipe using partial differential equations. Since available domain specific and general purpose behavioural simulators are only capable of solving ordinary differential equation systems, this approach has to be transformed (usually employing fixed discretisation of space).

These and other experiments carried out on the electronic simulators revealed the expected behaviour:

- In many cases the simulator did not find a solution at some point in time (the solver did not converge on some solution).
- Some cases terminated on solutions, but the results were not sufficient, since the detection and handling of discontinuous behaviour was not sufficient.

In this case, no experiments were carried out on long time transitional analysis.

5.2 What is a hard problem for a hydraulic simulator?

Having expressed this lacking qualities of analog circuit simulators for hydraulic problems, it should be mentioned again that the hydraulic solvers probably will have no proper strategies for complex electronic networks with about 10 000 transistors, due to a missing optimisation for this domain. We could not make such experiments as appropriate means for describing netlists of this size are still missing.

6 Conclusion

Within the automotive industry in Germany it has been decided to use VHDL-AMS as their unified modelling language for the description and exchange of physical and system components.

First industrial applications are carried out within the Brite/EuRam-Project *TOOLSYS*. Partners from the automotive industry and software makers exchange mechanic, hydraulic and controller models describing different mixed domain systems. Besides gaining knowledge on mixed domain modelling the partners will define requirements for mixed domain simulators.

First experiments have already shown that the unified language performs well for expressing the different domain models. But on the other hand, all simulators inspected seemed to be optimised for particular purposes and are not capable of mixed domain simulation, so far.

Unified modelling seems possible using VHDL-AMS. Mixed domain simulation with a unified simulator needs further development.

References

[1] K. Bakalar, E. Christen, and A. Vachoux. Analog and mixed-signal extensions to VHDL through examples. Technical report, IEEE DASC 1076.1, http://www.vhdl.org/analog/wwwpages/tutorial.html, 1997.

[2] I. Bausch-Gall. Modellschnittstellen zwischen Simulationsprogrammen. In A. Sydow, editor, *ASIM - Simulationstechnik, 8. Symposium*, 1993.

[3] I. Bausch-Gall. Eignung von VHDL-A zur Modellierung mechanischer und hydraulischer Systeme. Technical report, MSR, IAM, Postfach 2652, D-38016 Braunschweig, 1995.

[4] K. Brenan, S. Campbell, and L. Petzold. *Numerical Solutions of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.

[5] G. Grübel and C. P. Jobling. Simulation-tool abstractions based on the neutral dynamics-model description DSblock. In *IEE International Conference — Control*, 1994.

[6] Mitchell & Gauthier Associates. *ACSL Reference Manual*, 10.0 edition, 1991.

[7] N. Mittwollen. Hydraulic simulation of cavitation induced pressure fluctuations with peculiar periodicities in a fluid power unit. In Burrows and Edge, editors, *Design and Performance, 8th Bath International Fluid Power Workshop*, 1995.

[8] Abschlußdokumentation des Projekts MSR. Technical Report FZI-ESM-MSR-DO-01, MSR, http://www.msr-wg.de/, 1996.

[9] M. Otter. DSblock: A neutral description of dynamic systems. *Open CACSD Electronic Newsletter*, 1(3), 1992.

[10] C. Richards, D. Tilley, S. Tomlinson, and C. Burrows. Type-insensitive integration codes for the simulation of fluid power systems. In *Proceeding of the Winter Annual Meeting*, New York, 1990.

[11] K. Sanada, A. Kitagawa, and T. Takenada. A study on analytical methods by classification of column separations in a water pipeline. *Transactions of the Japanese Society of Mechanical Engineers, Series B*, 56(523):585–593, 1990.

[12] J. C. Strauss. The SCi continuous system simulation language (CSSL). *Simulation*, 9(6):281–304, 1967.

[13] A. Vachoux. VHDL-A: Analog and mixed-mode extension to VHDL. In *EUROSIM Simulation Congress*. EUROSIM, 1995.

[14] Definition of analog extensions to IEEE standard VHDL - draft. for working group purposes only, July 1997.