Framework and Multi-Formalism : the ASAR Project

P. Asar*
Correspondence should be sent to Michel Israel
LaMI Universite d'Evry, Bld. des Coquibus, 91025 Evry Cedex
email: israel@univ-evry.fr

Abstract

This paper describes a research project – named ASAR – grouping together six french research teams, oriented towards architectural and system synthesis. The main concern of this project is to build an hardware/software codesign framework.

Keywords: hardware/software codesign framework, multiple CAD tools integration, intermediate data flow graph.

1 Introduction

This paper describes a research project – named ASAR[1] – grouping together six french research teams . A main concern of this project is hardware/software codesign and user-interface management.

Most of these teams have been already involved in architectural synthesis for a long time, using distinct input formalisms and approaches. While independently developed, these approaches are complementary, and could be used during different stages of a design, or for designing different parts of a system.

The goal of the ASAR project is to build a generic architectural synthesis framework providing:

- the possibility of experimenting various approaches, corresponding to different target architectures and design description languages,
- multiple associated tools for transformations and synthesis.

This multi-formalism framework is currently under construction. It is built upon the CENTAUR [?] system, already used by three teams as a common generic software environment for user-interfaces, formalisms and tools management.

*P. Asar is a generic name for the ASAR project members, namely, M. Auguin (I3S), M. Belhadj (IRISA), J. Benzakki (LaMI), C. Carrière (I3S), G. Durrieu (CERT-Onera), Th. Gautier (IRISA-INRIA), M. Israël (LaMI), P. Le Guernic (IRISA-INRIA), M. Lemaître (CERT-Onera), E. Martin (LASTI), P. Quinton (IRISA-CNRS), L. Rideau (INRIA-Sophia), F. Rousseau (LaMI), O. Sentieys (LASTI).

[1] ASAR stands for "**A**telier d'accueil générique pour la **S**ynthèse **AR**chitecturale".

2 Framework structure

The overall structure of the ASAR framework is sketched in figure 1.

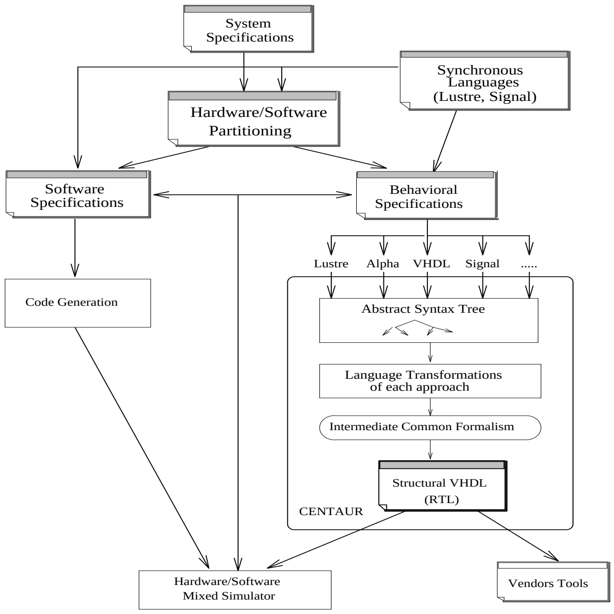

Figure 1: Overall structure of ASAR

2.1 Basis

Three components of ASAR are of main importance:

CENTAUR The CENTAUR generic environment is of course the first stone on which we can build this common framework, allowing to manipulate different syntaxes (abstract, concrete) and user interfaces in an uniform way.

An Internal Common Formalism (ICF) In order to federate our different approaches based on distinct user-formalisms, a sound semantic basis is needed, into which each formalism can be expressed. In other words, there is an underlying hidden common semantic to our multiple user-formalisms, to be conveyed by this ICF. This common formalism, a kind of common denominator, need not be human-readable, and could serve as a common exchange format between different tools.

As our different approaches are all synchronous, a synchronous data-flow semantic seems to be the most appropriate. It is the reason why the *GC* formalism is under investigation for the ICF. GC [?] is a common format for synchronous data-flow languages, developed in the french *projet SYNCHRONE*. Its use, as a ICF for ASAR, is under investigation.

VHDL The VHDL language, which is now a standard, cannot be ignored by any architecture oriented tool. Behavioral VHDL is of course one of the user-input language of ASAR, and structural VHDL will be one of its possible output format. VHDL is used by the OSYS [?] and GAUT [?] synthesis tools.

2.2 Multiple user-formalisms

In addition to VHDL, three other user-formalisms are present in ASAR, all belonging to the synchronous data-flow language family. SIGNAL [?] favours a relational expression, ALPHA [?] and LUSTRE [?] are purely functional languages, ALPHA being suited for regular and systolic architectures.

Translators from concrete to abstract syntaxes are already available under CENTAUR, for VHDL, SIGNAL, ALPHA and LUSTRE. Pretty-printers (from abstract to concrete syntaxes) also exist. Other translators are planned or could be integrated in ASAR via CENTAUR:

- translators from user-formalisms abstract syntax, to the ICF;
- translators from the ICF to the hardware domain: structural or synthesizable VHDL, EDIF; and to the software domain: C.

2.3 Multiple tools

The ASAR framework will provide a set of tools oriented towards particular target architectures or domains. GAUT is dedicated to signal processing designs. OSYS is a prototype of architectural synthesis system from behavioral VHDL descriptions. ALPHA and TRANSE [?] are tools based on a transformational interactive approach. The SIGNAL system oriented towards real-time systems design, provides proof checking tools and a SIGNAL to C compiler.

As said before, it is expected that different tools and formalisms will be used or at least experimented for the same design, using ASAR. In order to facilitate this multi-formalism approach, the role of the ICF appears essential:

- as a target for synthesizing and transforming tools: input-L $\rightarrow$ ICF,
- as a source for translators into languages, like structural VHDL or C, which could be inputs for other external tools: ICF $\rightarrow$ output-L,
- as source and target for possible new transforming tools: ICF $\rightarrow$ ICF.

3 Conclusion and future work

Three main objectives have been defined in this project:

- integrate the different architectural synthesis tools in a generic environment,

- define an internal common formalism (ICF),
- study the actual state of art in codesign in order to extract the different characteristics needed to be automated.

The last results relative to these points will be presented.

References

[1] J.-P. Paris, G. Berry, F. Mignard, Ph. Couronné, P. Caspi, N. Halbwachs, Y. Sorel, A. Benveniste, Th. Gautier, P. Le Guernic, F. Dupont, and C. Le Maire. *Projet SYNCHRONE – Les formats communs des langages synchrones.* Technical Report 157, INRIA, June 1993.

[2] P. Le Guernic, Th. Gautier, M. Le Borgne, and C. Le Maire. Programming real-time applications with SIGNAL. *Proceedings of the IEEE*, 79(9):1321–1336, September 1991.

[3] P. Borras, D. Clément, T. Despeyroux, J. Incerpi, G. Kahn, B. Lang, and V. Pascual. Centaur: the system. In *Proc. of ACM SIGSOFT '88: Third Symposium on Software Development Environments*, Boston, November 1988. (Also as INRIA research report Number 777).

[4] G. Durrieu and M. Lemaître. Design by transformation of synchronous descriptions. Submitted to the *3rd International Workshop on Algorithms and Parallel VLSI Architectures*, Leuven, Belgium, August 1994.

[5] N. Halbwachs. *Synchronous Programming of Reactive Systems.* Kluwer Academic Publishers, 1993.

[6] N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud. The synchronous dataflow programming language LUSTRE. *Proceedings of the IEEE*, 79(9):1305–1320, September 1991.

[7] P. ASAR. *Towards a Multi-formalism Framework for Architectural Synthesis : the ASAR Project* Codes/Cashes September 94, Grenoble, France.

[8] M. Israel and J. Benzakki. Osys: synthesis tool and hardware/software codesign. In *IFIP Codes/cashes93*, IFIP, Austria, May 1993.

[9] H. Le Verge, C. Mauras, and P. Quinton. The ALPHA language and its use for the design of systolic arrays. *Journal of VLSI Signal Processing*, 3:173–182, 1991.

[10] E. Martin, O. Sentieys, H. Dubois, and J.L. Philippe. An architectural synthesis tool for dedicated signal processors. In *EURO-DAC 93*, 1993.