The LARA Compiler Suite*

João Bispo[1], Pedro Pinto[1], Ricardo Nobre[1,2], Tiago Carvalho[1,2], João M. P. Cardoso[1,2]
[1]Departamento de Engenharia Informática
Faculdade de Engenharia (FEUP), Universidade do Porto, Portugal
[2]INESC-TEC, Porto Portugal
{jbispo, pmsp, ricardo.nobre, tiago.diogo.carvalho, jmpc}@fe.up.pt

Keywords—MATLAB, Source-to-Source Compilers, Aspect-Oriented Programming, LARA, Embedded Systems

LARA [1] is an aspect-oriented programming (AOP) language which allows the description of sophisticated code instrumentation schemes, advanced mapping strategies including conditional decisions, based on hardware/software resources, and of sophisticated sequences of compiler transformations. Furthermore, LARA provides mechanisms for controlling all elements of a toolchain in a consistent and systematic way, using a unified programming interface. We present three compiler tools developed around the LARA technology, MATISSE, MANET and ReflectC.

MATISSE [2, 3] is a compiler which 1) allows analyses and transformations on MATLAB code and 2) generates C code from the MATLAB code. MATISSE can be fully controlled through LARA aspects, which can define the type and shape of MATLAB variables, specify code insertion/removal actions, and define specialization directives and other additional information. MATISSE can output transformed MATLAB code and specialized C code. The knowledge provided by the LARA aspects allows MATISSE to generate C tailored to specific targets (e.g., use statically declared arrays to be compliant with the high-level synthesis tools such as *Catapult C*). The MATISSE compilation flow can be seen in Figure 1(a).

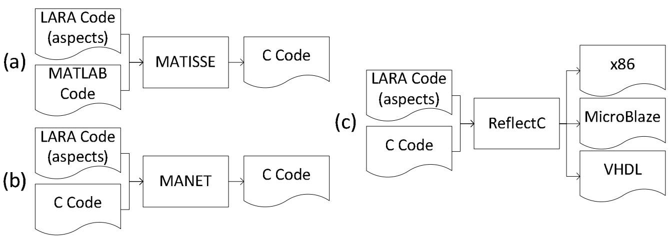

Figure 1. LARA-Guided Tools.

MANET [4] is a source-to-source compiler for ANSI C based on Cetus [5], and is controlled using LARA aspects. MANET manages to leverage the expressiveness and modularity of LARA to query and manipulate the Cetus AST, providing an easy compilation flow with main goal of code instrumentation and code transformations. This flow is depicted in Figure 1(b). LARA aspects allow for a simple selection of program elements in the code which can be analyzed or transformed, by either consulting their attributes or applying actions. Thus, MANET can be used to provide information reports based on compiler analyses, to implement sophisticated code instrumentation strategies, or to perform code optimizations and transformations.

ReflectC [6, 7] is a C compiler based on CoSy's compiler framework [8]. CoSy's configurability and retargetability make ReflectC particularly effective for exploration of compiler transformations and optimizations on possible architecture variations, and it is being used for hardware/software co-design and design space exploration (DSE). Figure 1(c) depicts the compilation flow of ReflectC, and its various target architectures.

We will present demos of the tools and the use of LARA aspects and strategies to guide our suite of compilation tools providing:

1) C code generation from MATLAB code, according to information provided by LARA aspects;
2) Instrumentation of C code to be used for collecting specific compile and runtime information (e.g., execution time, range of values for specific variables, custom profiling);
3) User-controlled compiler optimizations targeting several architectures and DSE of sequences of compiler optimizations bearing in mind performance improvements.

In addition to presenting examples for each of the tools of the LARA compilation suite, we show an execution of the complete toolchain, controlled by LARA aspects (see Figure 2).

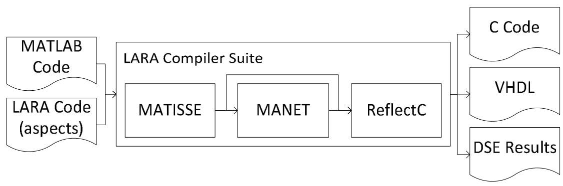

Figure 2. LARA Compiler Suite.

REFERENCES

[1] J.M.P. Cardoso, T. Carvalho, J.G. Coutinho, et al., "LARA: an aspect-oriented programming language for embedded systems," presented at the Proc. of the Int. Conf. on Aspect-oriented Software Development, AOSD'12, March, 2012.

[2] J. Bispo, P. Pinto, R. Nobre, et al., "The MATISSE MATLAB compiler," in 11th IEEE Int. Conf. on Industrial Informatics (INDIN), 2013, pp. 602-608.

[3] MATISSE, http://specs.fe.up.pt/tools/matisse/

[4] MANET, http://specs.fe.up.pt/tools/manet/

[5] C. Dave, H. Bae, S.-J. Min, et al., "Cetus: A source-to-source compiler infrastructure for multicores," Computer, 42(12), pp. 36-42, 2009.

[6] J.M. Cardoso, T. Carvalho, J.G. Coutinho, et al., "Controlling a Complete Hardware Synthesis Toolchain with LARA Aspects," Microprocessors and Microsystems, 2013.

[7] ReflectC, http://specs.fe.up.pt/tools/reflectc/

[8] ACE CoSy® Compiler Development System, http://www.ace.nl/compiler/cosy.html

*This work has been partially supported by FCT (Portuguese Science Foundation) under research grants PTDC/EIA-CCO/116796/2010, SFRH/BD/82606/2011, SFRH/BD/90507/2012, and FEDER/ON2 and FCT project NORTE-07-124-FEDER-000062. Previous support was provided by the European Community's Framework Programme 7 (FP7) under contract No. 248976 (REFLECT project).