Component Model Formalization and its Algorithms

Hélène Coullon and Christian Perez

LIP, ENS Lyon

Master Internship 2016

Keywords : Component-based software engineering, model formalization, algorithms, parallel patterns

Advisors : Hélène Coullon (AVALON) et Christian Perez (AVALON)

Laboratory : LIP, ENS Lyon, UMR CNRS-INRIA 5668, 46 allée d'Italie, 69364 Lyon Cedex 07

Phone : +33 (0)4 72 72 80 09

Email : helene.coullon@inria.fr, christian.perez@inria.fr

Context

Component-based Software Engineering [4] has proved many times good properties for code re-use, separation of concerns, maintainability and productivity of codes. As a matter of fact, those same properties are also needed in high performance parallel programming, where codes are difficult to produce for one given parallel hardware architecture (productivity problems), but even more difficult to maintain, to make generic or less hardware specific.

Component-based runtime models have already been proposed without performance overhead on high performance codes [1]. Those kind of models proposes very low level and simple concepts, as for example *primitive components* and method calls or MPI *connections* between them. However if low-level component models bring improvment in high performance codes, a higher-level language is needed to increase productivity and code-reuse. In this context the High Level Component Model (HLCM) has been proposed. In this model higher level concepts as *composite components*, *genericity* and *connectors* are proposed. From this higher level language a compilation phase produces a low-level high performance component-based program. This compilation phase is based on search, backtrack and reduction to primitive components.

A specialization calculus, called SpecMod [3], has recently been proposed. HLCM could be formalized using this calculus. This internship takes place in this context. First, a formalization of HLCM using SpecMod will be asked to the student, as well as an evaluation of this formalization through the definition of parallel algorithmic skeletons [2] (parallel patterns). Second, the student will have to write an algorithm for the HLCM compilation, which is linked to the formalization previously done, probably based on dynamic programming with search and backtrack.

Internship Objectives

The aim of the internship is :
— State of the art on component models, including L2C and HLCM, and on the SpecMod calculus. State of the art on algorithmic skeletons and parallel patterns.
— The SpecMod calculus will be used to encode the HLCM model.
— This formalization will be evaluated to define parallel algorithmic skeletons.
— From this formalization the compilation algorithm of HLCM, including a search and backtrack for good components and connectors specializations, will be proposed.
— Optionnaly, the student will have the opportunity to implement its new algorithm inside HLCM.

Comments

The internship will be at the LIP, ENS Lyon, Lyon

Références

[1] Julien Bigot and Christian Pérez. High Performance Composition Operators in Component Models. In Ian Foster, Wolfgang Gentzsch, Lucio Grandinetti, and Gerhard R. Joubert, editors, *High Performance Computing : From Grids and Clouds to Exascale*, volume 20 of *Advances in Parallel Computing*, pages 182 – 201. IOS Press, 2011.

[2] Murray I. Cole. *Algorithmic skeletons : a structured approach to the management of parallel computation*. PhD thesis, University of Glasgow, 1988. AAID-85022.

[3] Vincent Lanore and Christian Pérez. A Calculus Enabling Reuse and Composition of Component Assembly Specialization Processes. Research Report RR-8761, Inria, July 2015.

[4] Clemens Szyperski. *Component Software : Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.