

Title: High-level Low-level Transpilation Tools

Keywords: transpilation, high-level languages, C, IDE, type inference

Laboratoire, institution et université: INRIA Lille Nord Europe

Location: Lille, France

Team: [Equipe RMoD – INRIA Lille Nord Europe](#)

Internship supervisor:

Guillermo Polito guillermo.polito@univ-lille.fr

guillep.github.io

Context

Transpilation is a language implementation technique based in source-to-source translation. Instead of translating a program directly to machine code, work typically done by a compiler, a transpiler takes the source code of a programming language and translates it to another programming language, namely the target language. This technique allows to reuse the existing compilation chain of the target language, with its optimizations.

Slang is a Pharo-to-C transpiler aimed at generating efficient C code for writing critical applications such as virtual machines. To perform an efficient transpilation, the Slang transpiler accepts only a subset of Pharo source code and it applies several analyses and transformation to this subset: e.g., type inference, function inlining.

This transpilation has a key advantage: programs written in a Slang compatible subset can be also executed in Pharo, previous to its C translation. Slang developers do not need to wait until compilation is done to test or execute their code, and they can use high-level programming tools.

Objectives

The objective of this internship is to enhance the Slang transpilation toolchain to give better and faster feedback to developers. The student will work on the following topics

- fast incremental type inference to show developers typing errors before translation
- fast partial code analyses to detect non-translatable code before translation
- evaluate the need to augment the Slang transpiler with new type annotations or scoping information

References

Pharo: <https://pharo.org>

Back to the future: the story of Squeak, a practical Smalltalk written in itself.
Ingalls et al. OOPSLA '97

Two decades of smalltalk VM development: live VM development through simulation tools.
Miranda et al, VMIL '18