

# Implementing a Proof-Transforming Compiler from Eiffel to CIL

## PROJECT PLAN

Semester project  
Project period      WS 06/07  
Student name        Michel Guex  
Status (N-th semester) 9  
Email address       guexm@student.ethz.ch  
Supervisor name    Martin Nordio

## 1. PROJECT DESCRIPTION

### *Overview*

Goal of the semester project is to implement a proof-transforming compiler from Eiffel to .Net CIL. The source language is a subset of Eiffel, the logic for the subset of Eiffel is a Hoare style logic. The bytecode language is .Net CIL.

### *Scope of the work*

Design of an xml schema for the input proof in Eiffel and for the output proof in CIL  
Design and implementation of the API in Eiffel. The subset of the Eiffel language considered is: assign statement, compositional statement, if-then-else statement, loop statement. Optional: extending with rescue and retry statements.  
Implementation of the compiler: a parser for the input xml file and the actual translator

### *Intended results*

An xml file consisting of a proof of one class with several routines is given to the ptc. The output is the CIL proof written in a text file.

## 2. BACKGROUND MATERIAL

### *Reading list*

- Peter Müller, Martin Nordio, and Bertrand Meyer. A logic for Eiffel and its translation to .Net CIL.  
Peter Müller, Martin Nordio, and Bertrand Meyer. Proof-Transforming Compilation of Object-Oriented Programs to Bytecode.  
F. Y. Bannwart, A Logic for Bytecode and the translation of proof from sequential Java.  
A. Poetzsch-Heffter and P. Müller. A Programming Logic for Sequential Java.  
B. Meyer. Object-Oriented Software Construction.

### **3. PROJECT MANAGEMENT**

#### *Objectives and priorities*

Priorities are the API and the translator.

#### *Criteria for success*

The API, the translator and the parser have to be implemented for the statements for assignment, composition, if-then-else, and loop. Optional: extend with rescue and retry statements.

#### *Method of work*

Regular meetings are held with the supervisor.

### **4. PLAN WITH MILESTONES**

#### *Project steps*

- design of the API
- design of xml input file
- implementation of the compiler with the parser and the translator
- output to a text file

### **REFERENCES**

- [1] Chair of Software Engineering: *Semester-/Diplomarbeiten*; Online at: <http://se.inf.ethz.ch/projects/index.html>, consulted in October 2002.
- [2] Bertrand Meyer: *Object-Oriented Software Construction, 2nd edition*, Prentice Hall, 1997.