

# Features of C++ Template Metaprograms

**Norbert Pataki, József Mihalicza, Zalán Szűgyi,  
Viktor Májer, Zoltán Porkoláb**

Dept. of Programming Languages and Compilers,  
Fac. of Informatics, Eötvös Loránd University, Budapest  
e-mail:{patakino,pocok,lupin,majer\_v,gsd}@inf.elte.hu

## Abstract

*Template metaprogramming* is an emerging new direction in C++ programming for executing algorithms at compilation time in a functional way. With the assistant of template metaprogramming one can optimize runtime programs, emitting warnings and errors, implementing active libraries, developing domain-specific languages.

We present our framework, called Templight, which is able to debug code of C++ template metaprograms, and profile the compilation process. This framework is based on instrumentation, it extends the template code to emit warnings. We extract the trace from the warning messages. From the trace we can generate debugging and profiling data. We have also developed a graphical user interface to make the debugging process easier. In this application programmers can put down breakpoints, use step-in and step-out features, or continue the instantiations, etc.. We have measured the compilation time of some well-known template metaprogramming libraries to find bottlenecks in these codes.

Metaprograms run at compilation time, therefore we can add extra conditions when a program may be compiled. We present a solution for a typical object-oriented scantiness. Most object-oriented languages cannot declare members public from a given class but private from an other one. Eiffel supports selective visibility. Our solution takes advantage of power of template metaprogramming to make C++ more sophisticated.

*Keywords:* C++, template metaprogramming, debugger, profiler

*MSC:* 68N19 Other programming techniques

**Norbert Pataki, József Mihalicza, Zalán Szűgyi, Viktor Májer,  
Zoltán Porkoláb**

Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary