

---

**発表概要**

---

**オブジェクト指向言語に対するメタプログラミング言語の設計と実装**佐々木 卓<sup>†</sup> 亀山 幸義<sup>†</sup>

ソフトウェアの大規模化・複雑化に対応するため、オブジェクト指向言語とそれをさらに補強するさまざまな技術が研究されている。本研究は、それらの技術の1つであるメタプログラミングを、実用的なオブジェクト指向言語であるC++を対象として行うメタプログラミング言語PatternC++を提案する。メタプログラミングはある種のデータからソースコードを生成する技術であり、大規模ソフトウェアの効率的な作成に有効である。C++言語に対するメタプログラミングとしては、テンプレートを用いた技法(TMP)が一般に利用されている。しかし、テンプレートはメタプログラミングを意図した機能ではないため、記述が非常に煩雑となり、利用が難しいなどの問題点がある。さらに、オブジェクト指向プログラミングにおいては、パターンに基づくプログラムの設計、すなわち、個別のプログラム言語やデータ記述によらない形でのクラス設計が注目されている。パターンを利用して設計の本質を抽象することにより、設計の比較検討が容易になり、設計の再利用性が向上する等の利点がある。つまり、パターンを利用することによって複雑なプログラムを容易に扱うことができるようになる。本研究では、テンプレートの問題点を改善し、パターンに基づく開発を考慮に入れたメタプログラム言語を設計し実装を行った。

**The Design and an Implementaion of Pattern C++**SUGURU SASAKI<sup>†</sup> and YUKIYOSHI KAMEYAMA<sup>†</sup>

Various technologies that enhance the object oriented programming languages have been investigated by many researchers as we are facing increasingly large-scale and complex software. This paper addresses this issue, and in particular, proposes Pattern C++, a new metaprogramming language for the C++ language. Pattern C++ has two major advantages as a language; it improves over the existing method of metaprogramming for C++, and it makes it possible to design software in terms of patterns, that is, pattern-based design of programs. Metaprogramming is a method to automatically generate source codes from a certain set of data, and is thus effective to develop large-scale software. The existing, popular way to do metaprogramming for C++ is Template Metaprogramming (TMP). A big problem with TMP is that it is very difficult to use in practice. The reason we think is that templates are not intended to be a facility for metaprogramming, hence one needs a better language for metaprogramming in C++. Pattern-based program design is to design classes independently from any concrete representation of programs and data, and is by now one of the mainstream technologies in object oriented languages. Since the pattern-based design abstracts the essence of design, it becomes easy to compare one design with others, improve it, and reuse it. Therefore, pattern-based design enables one to deal easily with large, complex programs. In this paper, we introduce the design and an implementation of Pattern C++.

(平成19年3月22日発表)

---

<sup>†</sup> 筑波大学大学院システム情報工学研究科  
Graduate School of Systems and Information Engineering,  
University of Tsukuba