# 6 初心者にやさしいコンパイラ警告メッセージを目指して

応用

権藤克彦（東京工業大学）　内田公太（サイボウズ）
小島吉貴（元東京工業大学）　荒堀喜貴（東京工業大学）

Java や C などのプログラミング言語を初心者に教える際，まずは文法やアルゴリズムを重視することが多い．しかし，プログラミング経験として，実はコンパイラの警告メッセージも重要である．その理由は，GCC☆1 などのコンパイラのメッセージが初心者には理解が非常に難しいからである．

DrRacket など初心者向けの言語処理系もある．しかし，Java や C 言語など，プロも使用する「最初に学ぶ言語」の言語処理系で，初心者向けのものはほとんどない．そこで，ここではソフトウェア工学によるプログラミング理解と支援の一例として，コンパイラメッセージの調査と改善の研究[1),2)]を簡単に解説する．

## 現状：コンパイラのメッセージは初心者には理解が非常に難しい

プログラミングは本来はわくわくする知的な体験である．しかし GCC などのコンパイラの出力メッセージは初心者にはきわめて難解である．このため，初心者は「プログラミングは難しい」と感じやすい．これは GCC などは職業プログラマで対象で，初心者は非対象なことが原因である．

### ❖ GCC の悪いメッセージ例

たとえば，C 言語の `if(a==2&b==4)` というコードを考える．C 言語では `&` がビット AND，`&&` が論理 AND である．この例では，論理 AND のつもりで，間違ってビット AND を使ってしまっているとする．

このコード断片に対して，GCC-4.7.2 は以下の警告メッセージを出す．

```
warning: suggest parentheses around
comparison in operand of '&'
```

これは「`&` のオペランド中の比較周辺にカッコをつけた方が良いのでは？」を意味している．C 言語では `&` よりも `==` の方が優先順位が高い．このため，GCC はこのコード断片を `if((a==2)&(b==4))` と解釈している．そして，`if(a==(2&b)==4)` への変更を提案している．しかし，この警告メッセージは以下の点で初心者には非常に理解しにくい☆2．

- 演算子の優先順位の問題の可能性をコンパイラが指摘していることが，警告メッセージから読み取りにくい．
- 仮に読み取れたとしても，このコンパイラの指摘ではこの問題を解決できない（正しい修正結果は `if(a==2&&b==4)` である）．

## 既存コンパイラのメッセージ調査

既存コンパイラのメッセージ調査の研究として，初心者が間違えやすいコード例を評価したものがある[1)]．この研究では，初心者が間違えやすいコード例を 90 個収集し，それらを 9 つのコンパイラ・ツールに処理させ，4 つの指標（明瞭性，特定性，修正ヒント，平易な用語）で，エラーメッセージや警告メッセージを評価している．

さまざまな結果が得られているが，ここでは次の 2 つの主な結果を紹介する．

- コンパイラや静的解析ツールのメッセージのうち，

☆1 GNU Compiler Collection, https://gcc.gnu.org/

☆2 この警告メッセージは `if(x&0xFF00==0)` に対しては適切に働く．多くの場合，`if((x&0xFF00)==0)` がプログラマの意図だからである．

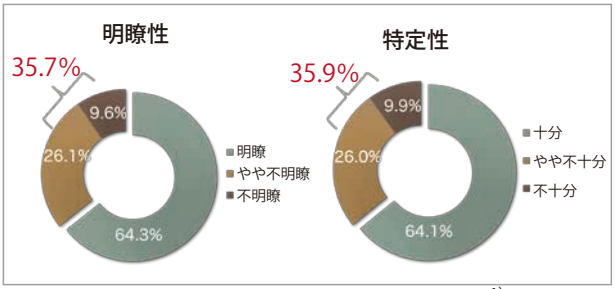


図 -1　約 1/3 のメッセージは明瞭性と特定性が悪い [1)]

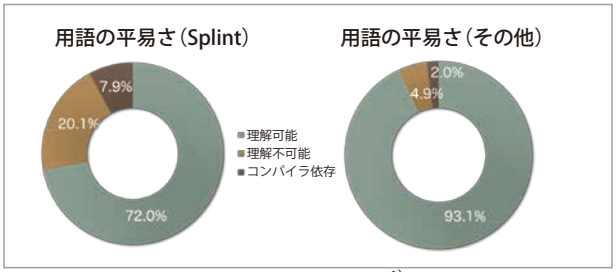


図 -2　Splint は用語の平易さが 4 倍悪い [1)]

約 1/3 は**明瞭性**（何が問題かはっきり分かるか）と**特定性**（問題を特定する具体的な情報はあるか）に欠ける（**図 -1**）．

- Splint はほかのコンパイラやツールよりも，**用語の平易さが約 4 倍悪い**（**図 -2**）．Splint は C 言語の静的解析ツールで，コンパイラが指摘しない怪しい部分も警告してくれる．これは Splint が積極的かつ正確なメッセージを出そうとして，皮肉にも初心者には難しい用語を多用したからと予想している（例：「副作用完了点」や「未定義動作」など言語規格の用語）．

また，Splint 以外のコンパイラやツールは**修正ヒントの出力が少なかった**．この結果は「初心者に分かりやすく，修正ヒントを多く出すことは難しい」ことを示唆している．

## C-Helper の試作

平易な用語で分かりやすいメッセージの出力を目指した C-Helper の試作も行われている [2)]．この試みでは，ヒューリスティクスを用いて 15 種類のエラーに対応している．たとえば**図 -3** では，`arr[20]="test.txt";` という代入に対して「char 型配列の 1 つの要素に文字列を格納できません．strcpy を使うことを検討してください」という分かりやすいメッセージを出力する．一方，GCC-4.7.2 は `warning: assignment makes intreger from pointer without a cast` という初心者には難しいメッセージを出力する．その一方で，C-Helper では不適切な警告メッセージの出力 (false positive) が増えるという悪い結果も得たと報告してる．

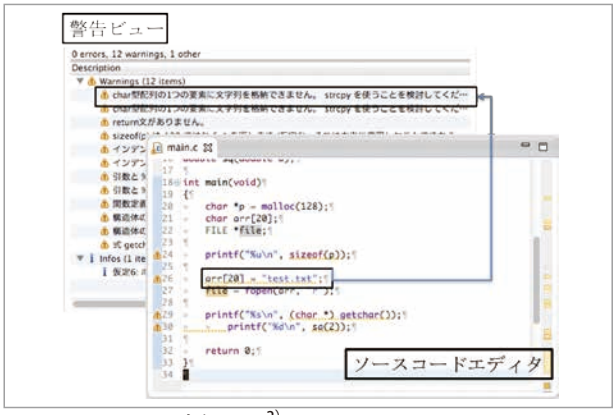


図 -3　C-Helper の実行画面 [2)]

C-Helper 以外にもコンパイラの警告メッセージの研究がある．たとえば，文献 3）は誤ったコード例，エラーメッセージ，修正したコード例をセットにして，カタログとして与えることで，エラーメッセージの理解を容易にする手法を提案している．これらのコンパイラの警告メッセージの研究は，**プログラマの意図は自動取得できない**ので本質的に難しい．しかし，この困難さ故に，これらの研究は面白いし，C-Helper などの研究により，この困難さへの挑戦が続けられている．

**参考文献**

1）Kojima, Y., Arahori, Y. and Gondow, K. : Investigating the Difficulty of Commercial-level Compiler Warning Messages for Novice Programmers, 7th Int. Conf. on Computer Supported Education(CSEDU 2015), pp.483-490 (2015).

2）Uchida, K. and Gondow, K. : C-Helper : C Latenterror Static/Heuristic Checker for Novice Programmers, 8th Int. Conf. on Computer Supported Education (CSEDU 2016), pp.321-329 (2016).

3）Kummerfeld, S. K. and Kay, J. : The Neglected Battle Elds of Syntax Errors, Proc. fifth Australasian conf. on Computing education - Vol.20 (ACE'03), pp.105-111 (2003).

（2017 年 7 月 19 日受付）

**権藤克彦**　gondow@cs.titech.ac.jp
**内田公太**　uchan0@gmail.com
**小島吉貴**　kojiyoshi09@yahoo.co.jp
**荒堀喜貴（正会員）**　arahori@cs.titech.ac.jp