6A-6

コーディング作法・ルールの品質特性による分類から見る傾向と効能

大野　克巳[†]　　平山　雅之[†]

[†]独立行政法人　情報処理推進機構(IPA)　ソフトウェア・エンジニアリング・センター(SEC)[‡]

１．はじめに

組込みソフトウェアの規模は拡大の一途を辿り、多人数による開発が主流となってきている。多人数による開発では多様なスキルレベルの開発者が参画するため、開発されるソフトウェアの品質にバラツキが生じやすくなる。このため開発者のスキルに依存しないような開発の標準化が求められている。特にＣ言語を利用したソフトウェアのコーディングでは、技術者のスキルによる影響が大きく、信頼性、保守性や移植性などを如何に適切なレベルに維持していくかが重要な課題となってきている。この課題の解決を目的として、従来から、開発者の暗黙の了解でコードを記述するのではなく、コーディング規約を形式知として定め、規約に基づいてコードを記述する手法が提案され利用されている。

SEC ではこうした従来のコーディング規約の考え方を一歩進め、ソースコードレベルでの信頼性/保守性/移植性を実現するためのコーディング作法の開発を進めている。本稿では我々が開発したこのコーディング作法（ESCR: Embedded System development Coding Reference）の概要とそれがソースコードの品質に与える影響について述べる。

２．ESCR の概要と特徴

ソースコードの標準化を目的としたコーディング規約については、MISRA-C など既に様々な規約が公開されている。しかし、これらの規約の多くは、経験に基づいて様々なルールが羅列されており、

①どのルールを守らなければいけないか

②どのルールはどの品質側面に影響があるか

といった情報が適切に整理されていない。また、既存のルールの多くは、ソースコードの読み易さ（可読性）などを重視したものが多く、このため信頼性や保守性など多様な品質側面を十分に網羅しておらず、総合的な品質特性達成という点では充分な効果が得にくくなっている。

このため IPA/SEC ではソフトウェアの品質特性を意識し、これらの特性を充足するという視点からコーディングルールを再整理し、『品質を保つために守るべきコードの書き方』としてそれらのルールの根底にある概念を作法として体系的に整理した。具体的に ESCR では、「JIS X 0129-1 ソフトウェア製品の品質第１部：品質モデル」さらに、それぞれの作法に C 言語に対応したルールをその必要性とともに提示し、コーディング作法ガイドとし提示している。ESCR では、JIS X0129-1 で示された６つの品質特性のうち信頼性・保守性・移植性・効率性が特にソースコードの品質に強く影響すると考え、これらの特性に対応する形でルールをカテゴライズしている。このガイドにおける作法、ルールの意味は以下の通りである。

作法　：ソースコードの品質を保つための慣習、実装の考え方であり、個々のルールの基本概念を示す。作法概要、作法詳細に階層化して示している。

ルール：守らなければならない、具体的な一つ一つの決め事であり、コーディング規約を構成する。

このガイドではリファレンスとして示している。なお、作法、ルールの多くは、複数の品質特性と関連する場合もあるが、最も関連の強い特性に分類している。（図１）

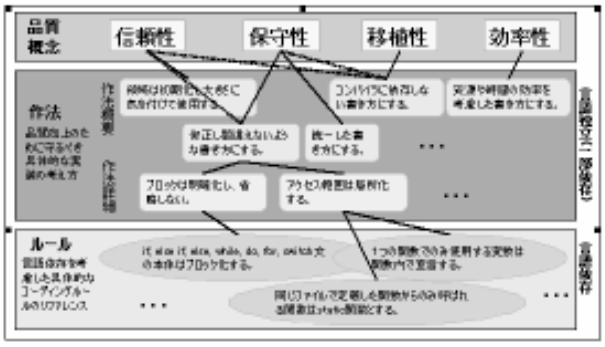

図１：品質概念・作法・ルールの関係

３．ESCR のソースコード品質への影響

３．１　品質への影響度

ここでは、ESCR で取り上げたコーディングルールのベースになっている“作法”と品質特性との関係から、ソースコード品質への影響について考察する。ESCR では参考として掲載しているコーディングルールを下記の３レベルに分けて扱っている。

レベル A:採否を条件によって選択してかまわないルール。

レベル B:常識として守るべきルール

レベル C:ルールとして絶対に遵守しなければいけないルール

このうえで図２は ESCR で規定している各品質特性別の”作法”毎にカテゴライズされるルールの数を整

An effect of "Coding practice and rule" watched from software quality characteristic

† Katsumi Ohno † Masayuki Hirayama

‡ Information-technology Promotion Agency, Japan (IPA) Software Engineering Center (SEC)

理したものである。この図を見ると ESPR の作法やサンプルルールを守ることで、ソースコードの品質特性について以下のような効果を期待することができる。

(a) 信頼性

「領域の初期化に関する作法」はレベル B が多く、コーディングの常識を身につけていない初心者への効果が期待できる。また「データの範囲・大きさに関する作法」は信頼性と極めて深い関係があり、結果としてルール数が多くなっており、これらを遵守することでコードレベルの信頼性向上を期待することができる。

(b) 保守性

保守性に関しては、「他人が読む事」「統一した書き方」の２つの作法に関して、特にレベル B,C の重要なルールが用意されている。その一方で、保守性に含まれる作法では、レベル A のルールの比率が比較的高く、製品に求められる保守性の度合いなどに応じて作法やルールを使い分けたほうが良いことを読み取ることができる。

(c) 移植性

移植性に関する作法のルールについては、その約半数がレベル A となっている。これは、製品で利用する対象の処理系を見据えて充分にルールを吟味する必要があることを示唆している。

このようにして品質特性別に ESPR の作法や付随する参照ルールをみると、信頼性、保守性、移植性などの特性によって、作法/ルールの遵守すべき度合いが若干異なっていることがわかる。このため最も手軽な方法として ESCR でカテゴライズしたレベル B,C のルールを最低限採用する方法も考えられるが、より進んだ利用方法としてプロジェクトや製品の特性を考慮してルールの取捨選択をする方法のほうがより大きな効果を得ることができると考えられる。

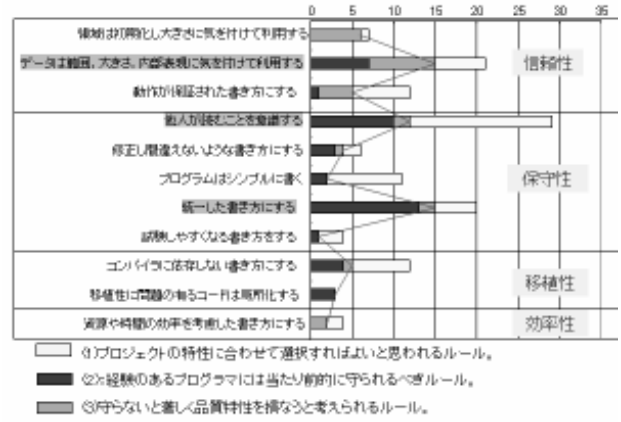

図２：コーディングルールの選択指針による分類

３．２　プロジェクト特性への影響度

一方、ESCR では実際のプロジェクトでのカスタマイズを意識し、参考として掲載しているルールを下記の３タイプに分類することもできる。

Type-1:プロジェクトごとにルールを規定し，文書化する必要があるルール

Type-2:複数のルールが提示されており，その中から選択する必要があるルール

Type-3:詳細を定める必要なし。そのまま規約として利用できるルール

図３は各作法ごとに整理された作法がどのタイプのものに属しているかを数値として整理したもので、Type-1,2 のルールが多く分布する作法はプロジェクト特性による影響度が高い作法と判断することができる。（図３）

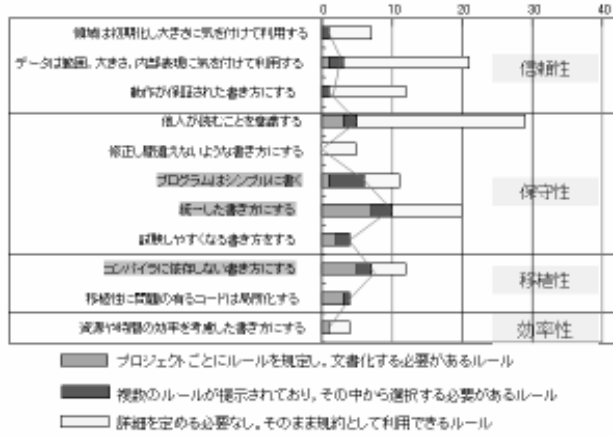

図３：コーディングルールの規約化指針による分類

このようにして見ると、信頼性・効率性に関連するルール群はプロジェクト特性による影響度に強い関連が見られず、ESCR のルールをそのまま使うことでこれらの特性をある程度保証することができると考えられる。一方、保守性に関しては「プログラムはシンプルに書く。」「統一した書き方にする。」といった作法でプロジェクト依存度が高くなっており、実際のプロジェクトの制約なども加味して利用する必要があることが分かる。

４．まとめ

本稿ではソフトウェアの品質特性を意識したコーディングの基本概念（作法）それに対応してコーディングルールの整理をした ESCR の概要を紹介し、またそれらがソースコードの品質に及ぼす影響について議論した。ESCR については既に IPA/SEC から公開され実用フェーズになっているため、今後、これらを実際に活用した場合の品質面への影響などについて実プロジェクトのデータをもとに追跡調査をしていく予定である。

参考文献

[1]組込みソフトウェア開発向けコーディング作法ガイド［C 言語版］ 2006 年　IPA/SEC

[2]組込みソフトウェア開発における品質向上の勧め（コーディング編）　2005 年　IPA/SEC