

# オブジェクト指向プログラムにおける インスタンス間関連の構造化ガイドライン\*

松原 義継<sup>††</sup> 山崎 直子<sup>†‡</sup> 掛下 哲郎<sup>†</sup>

†佐賀大学 理工学部 知能情報システム学科<sup>‡</sup>

‡有明工業高等専門学校<sup>§</sup>

3AE-5

## 1 まえがき

オブジェクト指向プログラムの複雑さを決める重要な要素としてオブジェクト間関連がある。これを構造化する概念として、Island[2]、複合関連[3]、Ensembles[4]等が様々提案されているが、これらをまとめてグループと呼ぶ。グループは、複数オブジェクトからなる集合を1つの高水準なオブジェクトとして抽象化したものである。従来のグループには、(1)複数のグループによるオブジェクトの共有を考えていない、(2)構造化による複雑さの低下を定量的に示していない、という問題点がある。

我々は、認知的心理学的アプローチを用いてソフトウェアの理解度(複雑さ)を計量するためのメトリクス[1]を提案している。本稿では、このメトリクスを用いてオブジェクト間関連に伴う複雑さを計量する。計量結果に基づいてオブジェクト間関連を構造化するためのガイドラインを導くことが本稿の目的である。メトリクス値を計算することでガイドラインの各項目間の重要度を定量的に示せるため、必ずしも強制を要しないものも列挙できる。

このメトリクスは、認知的心理学におけるチャンクと体制化の概念を利用している。チャンクは、何らかの意味で単一のまとまりを示すものに対応する。複数のチャンクから構成されるグループを1つの高水準チャンクにまとめて理解する行為を体制化と呼ぶ。 $n$ 個のチャンクからなるグループの理解コストは、以下のように与えられる。

$$C_R(n) = \begin{cases} n & (n \leq 7) \\ C_R(n-1) + (n-6) & (n \geq 7) \end{cases} \quad (1)$$

$n$ 個の構成チャンク  $u_i$  が内部構造を持っている場合、グループ  $U$  の理解コスト  $C(U)$  は以下のように与えられる。ここで  $C(u_i)$  は  $u_i$  の理解コストである。

$$C(U) = C_R(n) + \sum_{i=1}^n C(u_i) \quad (2)$$

## 2 オブジェクト間関連の複雑さ

始めに、構造化されたプログラムの複雑さを求める。ここで、メンバーがチャンクに対応する。メンバーは、

\*Structuring guideline for object-oriented program

<sup>†</sup>Yoshitsugu Matsubara, Naoko Yamasaki, Tetsuro Kakeshita

<sup>‡</sup>Department of Information Science, Faculty of Science and Engineering, Saga University, Saga 840, Japan.

<sup>§</sup>Ariake National College of Technology, Oomuta, Japan.

グループの構成要素に対応し、インスタンスもしくは(サブ)グループである。構造化を行わない場合、各インスタンスがメンバーである。 $n$ 個のインスタンスを  $m_1$ 個のグループで構造化し、更にこのグループを  $m_2, \dots, m_k$ 個と  $k(\geq 0)$ 段階に階層的に構造化する。この場合の関連の複雑さは式1,2より以下ようになる。

$$C(k) = C_R(m_k) + \sum_{i=1}^k m_i C_R\left(\frac{m_{i-1}}{m_i}\right) \quad (3)$$

ここで、 $m_0 = n, 1 < m_i < m_{i-1} (i = 1, \dots, k), m_{i-1} \bmod m_i = 0$ を満たす必要がある。第1項は、最上位レベルのグループ間関連による複雑さである。第2項は、各レベルでのグループ間関連に伴う複雑さであり、 $m_{i-1}/m_i$ はグループ当たりのメンバー数に対応する。 $k=0$ の時、この式は構造化を行わない場合の関連の複雑さを表す。

### 2.1 1レベルの構造化による複雑さの低減

1レベルの構造化とは、式3において  $k=1$ の場合である。もし、構造化により複雑さが減少するならば、以下の式が成立する。

$$C_R(n) > C_R(m_1) + m_1 C_R\left(\frac{n}{m_1}\right) \quad (4)$$

これを基に以下の性質を導くことができる。

1.  $n \leq 7$ の時、構造化は複雑さを増大させる。
2.  $n = n_0$ において、構造化により複雑さを低減できるならば、 $n \geq n_0$ となる全ての  $n$ において、構造化により複雑さが低減できる。

1の証明:  $n \leq 7$ より、 $C(0) = n, C(1) = m_1 + n$ なので、 $C(0) < C(1)$ となる。

2の証明: 仮定より、 $n = n_0$ において  $m'_1$ が存在して  $C_R(n_0) > C_R(m'_1) + m'_1 C_R\left(\frac{n_0}{m'_1}\right)$ が成立する。 $n_0 + 1$ 個のメンバーから構成される集合を  $m'_1 + 1$ 個のグループで構造化すると、関連の複雑さは以下の式で表される。

$$\begin{aligned} & C_R(m'_1 + 1) + m'_1 C_R\left(\frac{n_0}{m'_1}\right) + C_R(1) \\ & \leq C_R(m'_1) + m_1 - 4 + C_R\left(\frac{n_0}{m'_1}\right) \\ & < C_R(n_0) + m'_1 - 4 \\ & \leq C_R(n_0) + n_0 - 5 = C_R(n_0 + 1) \end{aligned}$$

従って、 $n = n_0 + 1$  の場合にも構造化により複雑さが低減できる。

以下に各種の  $n$  に対する最適な  $m_1$  の値を示す。

n	8	9	10	...	15	...	49
$C(0)$	9	12	16	...	51	...	952
$m_1$	2	2	2	...	3	...	4
$C(1)$	10	11	12	...	18	...	56

以上の考察に基づいて以下のガイドラインが得られる。

No.1 インスタンス数が8以下の場合、構造化を行わない方が良い。

## 2.2 複数レベルの構造化による複雑さの低減

$k$  段階の構造化を  $k+1$  段階にすることが望ましい場合には、 $C(k) > C(k+1)$  より以下の式が成立する。

$$C_R(m_k) > C_R(m_{k+1}) + m_{k+1} C_R\left(\frac{m_k}{m_{k+1}}\right) \quad (5)$$

式5は、式3において、 $n \rightarrow m_k, m_1 \rightarrow m_{k+1}$  と置き換えたものに対応するため、2.1節で示した2つの性質がそのまま成立する。

これらの性質を用いてメンバー数  $n$  のインスタンス集団に対する最適なグループを構成する。グループのメンバー数を7とし、そこから求まるグループを次のレベルで構造化する。これを、最上位レベルでのグループ数が7以下になるまで繰り返すと最適なグループを得ることができるため、以下のガイドラインが得られる。

No.2 グループ当たりのインスタンス数が7を越えないように構造化を行う。

## 2.3 共有メンバーの複雑さの性質

複数のグループに共有される共有メンバーは、グループ間の結合度を高め、複雑さを増大させる。そこで、この存在の複雑さへの影響を、先に用いたマトリクスを用いて示す。

メンバー数が  $n_i$  であるグループを考える。グループ内のメンバーが直接もしくは間接的に関連を持つメンバーの数を  $n_i^*$  とすると、関連数の複雑さは  $C_R(n_i^*)$  となる。共有メンバーが存在しなければ  $n_i^* = n_i$  である。グループ内の共有メンバーにより、 $t$  個のグループが直接もしくは間接的に結合していると、 $t$  個のグループの複雑さは

$$C_R(t) + \sum_{i=1}^t C_R(n_i^*) \quad (6)$$

となる。第1項は、共有メンバーにより結合したグループ間の関連を理解するコストである。第2項は、各グループの複雑さである。

共有メンバーから非共有メンバーへメッセージを送った場合、それは他のグループのメンバーへグループを経由せずメッセージを送ることになる。これは、グループ

を理解する際、送り先のメンバーの理解を必要とするため、グループのサイズを等価的に大きくする。1つのグループを理解するのに必要な総メンバー数は  $n_i^*$  であるが、この場合、 $n_i^* > n_i$  となる。

共有インスタンス同士でメッセージを送っても、 $n_i^*$  は変化しない。なぜなら、これはグループ内でのメッセージ通信だからである。共有されていないメンバー同士のメッセージ通信も、グループの複雑さには影響しない。式6に基づいて、以下のガイドラインが得られる。

No.3 共有メンバーにより直接もしくは間接的に関連を持つグループ数は少ない方が望ましい。

No.4 共有メンバーから共有されていないメンバーへはメッセージを流さない方が望ましい。

## 3 まとめ

本稿では、オブジェクト指向プログラムにおけるインスタンス間関連の複雑さを定式化し、これを低減するための構造化ガイドラインを提案した。ガイドラインに反した構造化を行った場合、複雑さがどのように増大するかを以下に列挙する。

No.1 構造化を行わない場合より、グループ数に比例して複雑さが増大する。

No.2 メンバーの複雑さが7を越えた場合、メンバー数の2乗に比例して複雑さが増大する。

No.3 結合したグループ数が7を越えた場合、グループ数の2乗に比例して複雑さが増大する。

No.4 メッセージを送ったメンバー数が、グループのメンバー数に加算される。

これらの内容はマトリクスにより計量ができるので、当該プログラムの状況に応じて計量した結果を基に重要度を付けることができる。現在、本ガイドラインをC++プログラムに対してチェックするソフトウェアツールを設計中である。

## 参考文献

- [1] 掛下, 松原, 山崎 " 認知心理学的アプローチに基づいたオブジェクト指向設計の複雑さ計量法 ", 日本ソフトウェア科学会 FOSE'96, 1996.
- [2] J. Hogg, " Islands: Aliasing Protection In Object-Oriented Languages ", OOPSLA, pp. 271-285, 1991.
- [3] B. B. Kristensen, " Complex associations: abstractions in object-oriented modeling ", OOPSLA, pp. 272-286, 1994.
- [4] Dennis de Champeaux, " Object-Oriented Analysis and Top-Down Software Development ", ECOOP, pp. 360-375, 1991.