フラクタル次元に基づくソフトウェア評価法に関する研究

4L−5

佐藤　和寿　　沢村　浩　　篠崎　明　　伊與田　光宏
千葉工業大学

1．初めに

フラクタル理論は、B.B.Mandelbrotによって" The Fractal Geometry of Nature" 【1】内で提唱された。同理論が広く認められるようになったのは最近である。現在、カオス的な要素を含むシミュレーションや様々な自然現象のモデリング、コンピュータグラフィックにおけるテクスチャマッピング、符号法など、その応用は多岐に渡る。同理論は未だ発展途上の概念であるが、まだ多くの可能性が秘められていると思われる。本研究では特にフラクタル次元が構造の複雑性を表すことに注目している。複雑性は、定義が曖昧なため一般的な尺度が存在しない。複雑性に単位を必要とする事象が存在するなら、フラクタル理論に基づき評価の指針とする事も十分な価値があるであろう。

2．目的

本研究の目的は、本研究では複雑性と密接な関係がある同理論をソフトウェア工学の分野に応用し、妥当性、有効性を検討していく。ソフトウェア品質の分析において現在様々な評価法が提案されている。ソフトウェアの持つ複雑性は様々な要因が絡み合い、一意的には分析できないと考える。本研究の手法は既存の方法と異なる側面からの分析によって、複雑性をさらに具体的に分析する一手段として位置づけられる。

3．フラクタル理論の概要

フラクタル次元は既知の整数次元の間に存在する。1次元の構造、すなわち直線を考える。これを2倍したとき、その長さは2倍になる。同様に、2次元の平面、3次元の立体を2倍に拡大したとき、それぞれ4倍の面積、8倍の体積になる。図１のように各構造をD次元としたとき、尺度を2倍にすると構造の大きさは2^D倍になる（図１）。

この際、各次元の測度の倍率は、

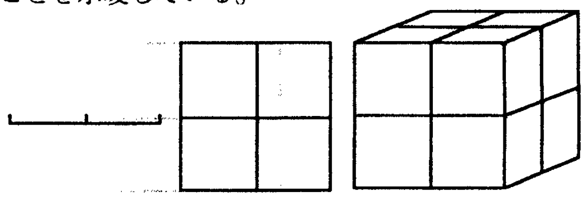
1次元　$2 = 2 = 2^1$
2次元　$4 = 4 = 2^2$
3次元　$8 = 8 = 2^3$

となる。D次元の構造をａ倍した時、測度がｂ倍になるとする。一般式は、

$$b = a^D \qquad (式1)$$

$$D = \log_a b \qquad (式2)$$

と表せる。対象がフラクタル構造の時、Dは非整数を取る。フラクタル次元は主に構造の複雑性を定量的に表現していて、複雑な構造ほど大きい値をとる。例えば、入り組んだ曲線のフラクタル次元は1以上2未満の実数で表される。これは、直線（１次元）の構造より複雑ではあるが、平面（２次元）を多い尽くすほど入り組んだ構造ではないことを示唆している。



2^1倍　2^2倍　2^3倍

図１　各次元の構造例

4．ソフトウエア工学の概要

ソフトウェア工学とはソフトウェアを計画、分析、設計、製造、テスト、保守を必要とする工学的製品（engineered product）として取り扱う為のものである。目的は、高品質なソフトウェアを正当なコストで効率よく生産、運用、保守することにある。ソフトウェア工学において定量的尺度を用いるのは、的確な判断のため、また工学的アプローチのために最も重要な課題である。しかし、定量的尺度は算出が困難で、品質や分量のごくわずかな部分にしか存在しない。システムや個々のモジュールにおいて、複雑性から受ける影響は大きい。複雑性は、設計、開発、保守等、多くの段階で計測されるべき尺度であるが、残念ながら現在までに様々な研究の報告があるものの、すべての尺度を規定するには至っていない。

An Evaluation of Software Modules
based on Fractal Dimension
Kazutoshi SATOH、Hiroshi SAWAMURA
AKIRA Shinozaki、Mitsuhito IYODA
Chiba Institute of Technology

5. ソフトウェア解析法

フラクタル次元は構造の複雑性を定量的に表す事ができる。本研究では、ソフトウェアの制御構造に着目しソフトウェアの品質評価に応用することを考える。まず、C言語のソースプログラムより、制御の流れに関する情報を抽出する。この際、モジュールにはフィルターをかけ、一定のフォーマットに整形する。次に抽出した構造に対して、一辺がrの格子（セル）に分割し、それを一部でも含むセルの数N(r)をカウントする（図5-1）。基本となるセルの一辺の長さをr_1、A倍に拡大したセルの一辺の長さをr_2、それぞれの時カウントされたセルの数を$N(r_1)$、$N(r_2)$とすれば、（式1）（式2）より次式を得る。

$$\frac{N(r_2)}{N(r_1)} = A^D \qquad \text{（式3）}$$

$$D = \log_A \frac{N(r_2)}{N(r_1)} \qquad \text{（式4）}$$

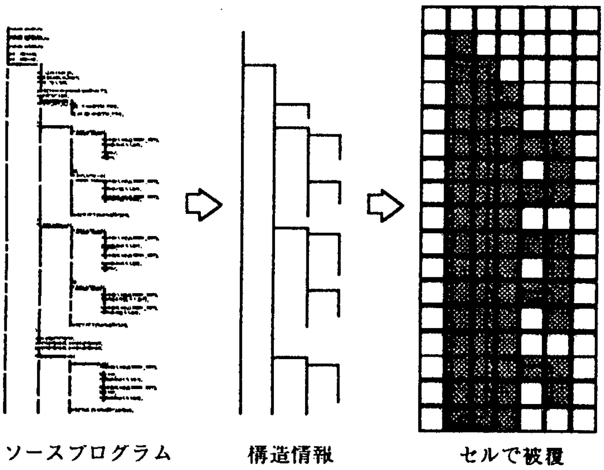


図2　構造情報の被覆

6. 測定及び結果と考察

C言語のモジュールに対してフラクタル次元を測定した。始めに測定を行ったモジュールは、THINK C / Symantec C++ 7.0 に添付の標準ライブラリのprintf.cを対象とした。被覆するセルの一辺の長さr（以下、r）を1:2から1:9まで変化させて測定した。ところが、安定した結果が得られずフラクタル次元が収束するのかどうか判断が困難であった。これは対象が、厳密に定義されたフラクタル構造でないことに起因すると考える。測定値が一つの値に収束しない事、また観測の尺度が測定値に著しい影響を与えてしまう事は評価法として適切とは言いがたい。そこで、rとN(r)の関係をグラフ化した（図3、4）。N(r)はr^{-D}に比例する。両図においてBはN(r)の実測値の推移、また、A、Cはそれぞれ対象の構造が1次元、2次元と仮定した場合のN(r)の推移である。

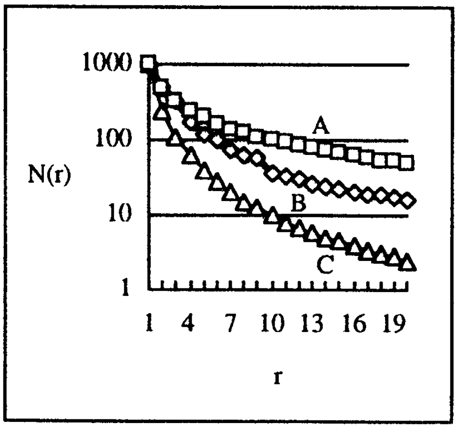


図3　printf.cの測定結果

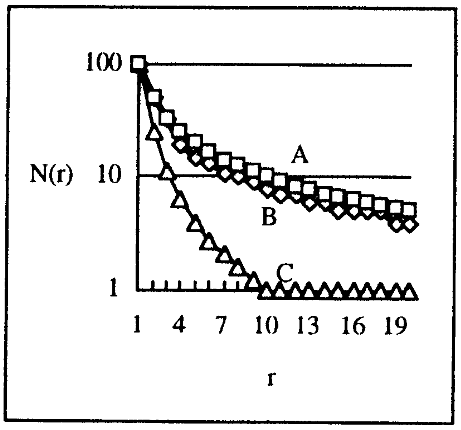


図4　stdio.hの測定結果

7. 終わりに

stdio.hはヘッダファイルなので、その記述の構造自体はprintf.cと比較すると単純である。図3のBは、図4に比べるとAに接近しており1次元に近いことが分かり、printf.cがstdio.hより複雑性の高いことがグラフにも反映されている。しかし、対象とするモジュールによって評価が安定しない傾向がある等の問題点も残している。

参考文献

【1】B. B. Mandelbrot、The Fractal Geometry of Nature（Freeman、SanFrancisco、1982）；広中平祐監訳、フラクタル幾何学　（日系サイエンス社、1985）