

## 特性関数インタフェースを用いた工数見積りツールの試作

5 S - 3

大瀧 陽悦, 吉田 奈穂子, 乗貞 雅, 岡田 正志

日本電気ソフトウェア(株)

## 1. はじめに

現在の業務システムのソフトウェア開発での初期段階の工数見積り作業は、規模を見積った後、既に得られている生産性の基準値に対し、新規性や難易度等種々の変動要因の変動比率によりモディファイした基準値を用いて、開発工数の見積り値を求めているのが一般的である。

このモディフィケーションでは、それぞれの変動要因毎に生産性の基準値を0. X~1. Y倍している。この過程では見積りを行う者はこれらの変動比率を厳密には確信していない場合が大部分である。

そこでこの過程にファジ理論で用いている特性関数を利用するツールを試作したので、ツールの概要と評価結果について報告する。

## 2. 工数見積り作業の実際

本ツールでは

- ① バッチ
- ② オンライン
- ③ 端末アプリケーション

と分けることができる業務システムを対象とした。この場合初期段階での工数見積り作業は次の手順である。

- (1) ①~③の各種別毎のプログラム本数の見積り
  - (2) 各種別の変動要因(新規性や難易度等)ごとに変動比率の決定
  - (3) ①~③の各種別毎の標準ステップ数と(2)によりモディファイした生産性を用いての開発工数の計算
- (1)~(3)の結果の開発工数算出式を示すと、各プログラム種別毎次のようになる。

$$N \times S / (P \Pi A_i)$$

N : 見積りプログラム本数  
S : 標準ステップ数  
P : 標準生産性

ここでの各変動要因の変動比率 $A_i$ は、それぞれ0. Xとか1. Yとか孤立した値で与えられる。

## 3. 試作ツールの概要

2. で述べた従来の見積り方式の基準値モディフィケーションでは、基準値を0. X~1. Y倍している。例えば、ある変動要因の変動比率を0. 8とした場合、これまでは強い確信を持てる0. 8も余り確信を持ってない0. 8も同じ0. 8として計算している。

そこで2の(1)から(3)に対応する作業をファジ理論を用いたルールベースのエキスパートシステムで、上記0. X倍とか1. Y倍とかを特性関数で入力できるツールを試作した。これにより、利用者は変動要因の変動比率としていただいているイメージに近い値をツールに与えることができるようになる。

基準値の変動要因は、確立したものがあるわけではなく各部門で独自に決めているのが現状である。要因が多くなれば見積りの精度は向上するかもしれないがそれだけツールの使い勝手も複雑になる。今回は最初の試作ということで基準値の変動要因として次の3つ

新規性 スキル 難易度

を用いてツールを試作した。

利用者からみたツールの使い方は次のようになる：

- (1) 予めバッチ/オンライン/端末アプリケーション別の標準ステップ数、生産性をツールに入れる
- (2) これから構築しようとするシステムのバッチ/オンライン/端末アプリケーション別の本数とそれぞれについて新規性/スキル/難易度の変動比率を0. X倍とか1. Y倍とか孤立した値ではなく特性関数で与える

これらをツールに与えると利用者は構築しようとするシステムの開発工数を得ることができる。ここで、(2)は構築しようとするシステム毎に与えるが、(1)は標準値を変更する時のみツールに設定する。

## 4. 評価

今回は先ず次の項目について評価した：

- ① ルール数による結果の影響
- ② ルール条件部の特性関数の形による結果の影響
- ③ ルール条件部の特性関数の定義域の分割数による結果の影響

## 4. 1 ルール数による結果の影響

基準値の変動要因として次の3つ

新規性 スキル 難易度

を選んだのでルールの形は

もし 新規性 が XXX で かつ  
スキル が XXX で かつ  
難易度 が XXX

ならば

生産性は XXX である  
 となる。ここで XXX には  
 高い 少し高い 普通 少し低い 低い  
 の5種類の値を取らせた。

ルールが13個の場合と85個の場合について8種類の業務システムのパターンを実行した結果が図1である。システムのパターンを指定する場合、今回はまだ自由な特性関数は指定できず、ルールで用いているのと同じ特性関数のみ指定可として評価を進めた。図での横軸は  
 高い 少し高い 普通 少し低い 低い  
 にそれぞれ

0.8 0.9 1.0 1.2 1.4

の値を持たせ（但しスキルについてはこの逆順）、各システムのパターンの重み付け値を掛け合わせた値である。

図1から少なくとも13個のルールでは不十分なことがわかる。

4.2 ルール条件部の特性関数の形による結果の影響

特性関数の傾きによる結果の違いをみるために2種類の特性関数を用意し、実行した。用いた特性関数を図2に、結果を図3に示す。図3の横軸の見方は図1と同じである。重み付け見積りは、従来方式の見積りである。

特性関数の傾きを鋭角にした方が従来方式の見積りの結果に近くなっている。

4.3 ルール条件部の特性関数の定義域の分割数による結果の影響

ファジィ推論を進めるためには特性関数の交点を求める必要がある。ここでの推論ではその交点を厳密な値として求めているわけではなく、特性関数の定義域を均等に分割し、交点に最も近い分割点を疑似的に交点として推論を進めている。そこでどの位の分割数が適切かみる必要があり、20等分と40等分に分割し実行した。その結果を図4に示す。

20等分と40等分で顕著な違いはみられないので、20等分でもほぼ問題ない。

5. まとめ

ここでは生産性の基準値をある定めた変動要因によりモディファイするときに、特性関数のインタフェースを用いるツールの評価結果について述べた。モディフィケーションに特性関数のインタフェースを用いた方がユーザフレンドリなツールとなることは明白なことから、その場合にツールがどの位複雑でなければならないかを評価の主眼に置いた。

本ツールによりソフトウェア開発の見積りが解決するわけではなく、基準値やその変動要因を明らかにすることの方が重要である。そのためには正確な開発データを収集することが重要であるが、まだ収集の方式についても議論されているところである（参考文献1）。

ここで述べた方式を盛り込んだソフトウェアの管理支

援ツールを作成・運用し、見積り作業を効果的に進めていく予定である。

参考文献

[1] 村田 他：工程管理ツールにおける一考察，情報処理学会第44回全国大会論文集

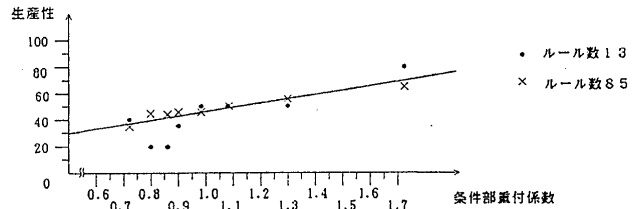


図1 ルール数13個と85個の生産性比較

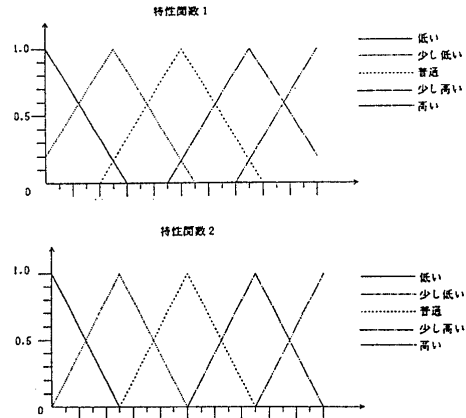


図2 特性関数1・2

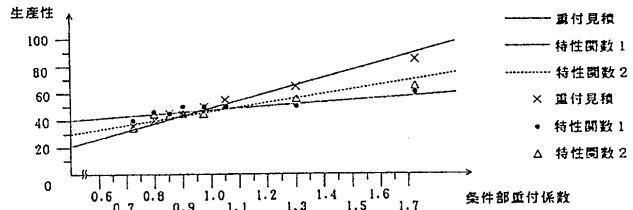


図3 特性関数1・2の生産性比較

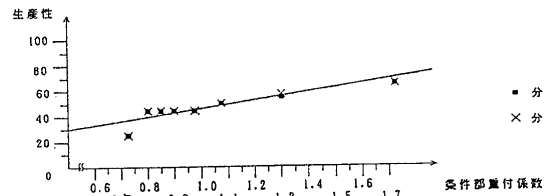


図4 特性関数定義域分割数20・40での生産性比較