

ソフトウェアプロジェクトシミュレータのための パラメータ値の設定

水野 修, 亀山 忠秀, 楠本 真二, 菊野 亨
大阪大学 大学院基礎工学研究科 情報数理系専攻
〒 560-8531 大阪府豊中市待兼山町 1-3
Tel: 06-850-6567 Fax: 06-850-6569
E-mail: o-mizuno@ics.es.osaka-u.ac.jp

要旨

本研究では, 一般化確率ペトリネット (GSPN) に基づくソフトウェア開発プロジェクトシミュレータのためのパラメータ値の設定法の提案と, その適用実験について述べる.

プロジェクトシミュレータにおけるトランジションの発火レートを定めるパラメータは, プロジェクト毎に一般には異なる作業環境などを反映するためのものである. これまでのシミュレーションでは特定のプロジェクトに特化したパラメータ値を他の類似のプロジェクトにも使用してきた. シミュレータ適用を一般化するにはこのパラメータ値の設定を自動化する必要があった.

ここでは設計, コーディング, レビュー, テスト, デバッグの各作業モデルの発火レートの定義中に含まれるパラメータ K_{cm} , K_{th} , K_{wr} の値と混入フォールト数の定義中に含まれるパラメータ K_{in} の値を各プロジェクト毎に求める手順を提案する. 更に, その適用可能性を確認するために行ったシミュレーション実験について述べる. なお, 本実験では企業から提供を受けた実際の開発プロジェクトのデータを利用した. 実験結果より, 提案するパラメータ値の設定法の有効性が限定的ではあるが確認できた.

1 研究の背景

我々はこれまでに一般化確率ペトリネット (GSPN)[1]に基づいたソフトウェア開発プロジェクトシミュレータの開発を行ってきた [7][8]. このシミュレータの開発に当たっては, 開発現場で収集可能な (あるいは既に利用されている) データをモデル上のパラメータとして可能な限り取り入れることを心がけた.

シミュレータでは開発プロセスの設計, コーディング, レビュー, テスト, デバッグの各作業を GSPN を用いてモデル化した. GSPN 上のトランジションが作業者の行動 (作業者間のコミュニケーション, 思考, 記述) や作業中に発生する事象 (フォールトの混入) に対応している. シミュレータでは作業量という概念を導入して, 発生する動的な事象をモデル上で再現できるようにしている. 作業間に存在する実行上の前後関係をコントロールフローを用いて記述し, 各作業に割り当てる作業員や, 入力となるプロダクト, 消費すべき作業量などを与えると 1 つのプロジェクトの記述が完成する.

このソフトウェアプロジェクトシミュレータは既に UNIX 上で実装しており, ユーザーインターフェイスには GUI を用いて直観的な使用が可能となるようにしている. これまでにこのシミュレータを用いた幾つかの適用実験を行い, 品質予測への適用可能性について発表してきている [9][10].

本シミュレータを動かすには, 各作業モデルの発火レートの定義中に含まれるパラメータと混入フォールト数の定義中に含まれるパラメータの値の設定を行わなければならない. パラメータは相互に関連しているため複数のパラメータの値を同時に設定する必要があり, その作業は非常に難しい. これまでの適用実験 [7][8][9][10] では, あるプロジェクトに特化したパラメータの値を他の類似のプロジェクトにも利用してきた. これでも不都合を生じなかったのは, 同一組織内では作業者の特性やプロジェクトの進行状況はそれほど変化しないためであると考えられる. しかし, 特性の異なるプロジェクトに対してもプロジェクトシミュレータの適用を可能とするためには, 与えられたプロジェクトに対してパラメータの値を求める一般的な手順を確立することが必要となる.

本研究では、シミュレータ実行のために必要となるペトリネットモデル中のパラメータの値を各プロジェクト毎に設定する方法を提案する。具体的には、設計、コーディング、レビューテスト、デバッグの各作業モデルの発火レートの定義中に含まれるパラメータ K_{cm} 、 K_{th} 、 K_{wr} の値と混入フォールト数の定義中に含まれるパラメータ K_{in} の値を各プロジェクト毎に求める手順を与える。更に、その適用可能性を確認するために行ったシミュレーション実験についても述べる。なお本実験ではある企業から提供を受けた2組の実際の開発プロジェクト PR_1 、 PR_2 のデータを利用した。その実験結果より、提案するパラメータの設定方法の有効性が限定的ではあるが確認できたものと考えている。

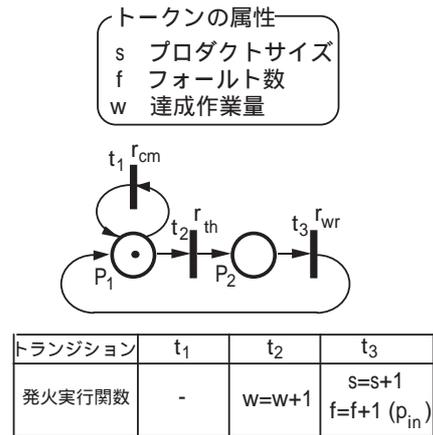


図 1: 設計作業の記述例

2 シミュレータの概要

まず初めに、我々が開発している一般化確率ペトリネットに基づくソフトウェアプロジェクトシミュレータについて述べる。

2.1 作業モデル

ここでは各作業をモデル化する方法について説明する。作業モデルは作業の種類毎に定義し、その記述には拡張 GSPN を用いる。作業の種類として設計、コーディング、レビュー、テスト、デバッグの5つを考える。

図 1は拡張 GSPN で記述した設計作業である。この記述ではトークンの属性としてプロダクトサイズ s 、フォールト数 f 、達成作業量 w の3つを持たせている。各属性の意味は次の通りである。(1) プロダクトサイズ s は作業により生成されたプロダクトのサイズを、(2) フォールト数 f は作業中に作り込まれたり、取り除かれたフォールトの個数を、(3) 達成作業量 w はその時点までに達成された作業量を、それぞれ表す。

図 1中のトランジションはすべて発火に時間を要する時間トランジション(図中の太線)である。例えば、トランジション t_1 の発火レートは r_{cm} で与えられており、これはトランジション t_1 の発火遅延時間が平均 $1/r_{cm}$ の指数分布に従うことを意味する [1]。これらのトランジションには思考、記述、会話といった作業者の振る舞い、あるいは、テスト中での故障の発生といった作業中の事象を対応させる。一方、ブレースは振る舞いや事象が起きるまでの待ち状態を表している。

図 1に示した設計作業の場合、トランジション t_1, t_2, t_3 はそれぞれ作業者間のコミュニケーション、問題の解決に必要な思考、ドキュメントとして記録す

るための記述に対応しており、発火レートとしてそれぞれ r_{cm}, r_{th}, r_{wr} を与えている。

2.2 発火レートとフォールト混入率

発火レートは一般に次のような関数として表す。

コミュニケーションレート

$$r_{cm} = f_{cm}(\text{作業者数, 各作業者の経験レベル, 入力プロダクトの完成度})$$

思考レート

$$r_{th} = f_{th}(\text{作業者数, 各作業者の経験レベル})$$

記述レート

$$r_{wr} = f_{wr}(\text{作業者数, 各作業者の経験レベル})$$

これらの関数の利用により作業者数や経験レベル、あるいは、入力プロダクトの完成度に応じて、コミュニケーションの発生頻度、思考や記述の難易度を動的に設定することができる

また、思考や記述のトランジションが発火する毎に達成作業量 w やプロダクトサイズ s が増加するといったトークンの属性値の変化は図 1の記述にもある様に、各トランジションの発火実行関数として記述される。

なお、設計作業におけるフォールトの混入(すなわち、フォールト数 f の変化)はフォールト混入率 p_{in} による確率的な事象として取り扱う。このフォールト混入率 p_{in} は一般に次のような関数として表す。

$$p_{in} = f_{in}(\text{作業者数, 各作業者の経験レベル, 予定終了期日, 入力プロダクトの完成度})$$

この関数の利用により、デッドラインや作業者の経験がフォールトの混入率に与える影響も考慮すること

ができる．例えば，図1の設計作業ではトークンが P_1 にあるとき，2つのトランジション t_1 (コミュニケーションを表す) と t_2 (思考を表す) が発火可能である． t_1 の発火によって時間は経過するがプロダクトや作業量の達成には何の影響もなく，トークンは P_1 に戻る．次に， t_2 が発火すると発火実行関数によって達成作業量 w が1増加するとともにトークンは P_2 に移る．トークンが P_2 にあるときは t_3 だけが発火可能である．この t_3 (記述を表す) が発火するとプロダクトサイズ s が増加し，フォールト混入率 p_{in} の値によってフォールト数 f の増加が確率的に起こる．このときトークンは P_1 に移る．

2.3 パラメータ K_{cm} , K_{th} , K_{wr} , K_{in}

今回のシミュレータの開発では f_{cm} , f_{th} , f_{wr} , f_{in} のそれぞれの関数を次のように定義している．以下では M は作業員数， L は作業員の経験レベルの和， R は入力プロダクトの完成度， D は予定終了期日までの日数である．なお， K_{cm} , K_{th} , K_{wr} , K_{in} はパラメータであって一般には作業毎に異なる値が与えられる．

$$r_{cm} = K_{cm} \times \frac{M^2}{L \times R}$$

$$r_{th} = K_{th} \times \frac{L}{M} \times M = K_{th} \times L$$

$$r_{wr} = K_{wr} \times \frac{L}{M} \times M = K_{wr} \times L$$

$$p_{in} = K_{in} \times \frac{M}{L \times R \times D} \times M$$

この定式化について説明する．コミュニケーションレート r_{cm} は作業員数の2乗に比例し，作業員の経験レベルと入力プロダクトの完成度に反比例する．また，思考レート r_{th} と記述レート r_{wr} は共に作業員の経験レベルに比例する．さらに，混入フォールト率 p_{in} は作業員数に比例し，各作業員の経験レベル，入力プロダクトの完成度，締切までの日数に反比例する．この定式化は文献 [3][4][11] や，作業員へのインタビューなどに基づいて行っている．

3 パラメータ値の設定

2.3節で説明した発火レートの式に含まれるパラメータの K_{cm} , K_{th} , K_{wr} 等を発火レートパラメータと呼ぶことにする．ここでは，この発火レートパラメータの値を設定する方法について述べる．

3.1 設定手順の概要

プロジェクトの各作業毎にそれが実際に行われた状況を忠実に再現した動きを GSPN がするように，パラメータを設定する．具体的には，実際に完了しているプロジェクト PR のデータから得られる作業員数，作業時間などを作業モデルに与え，シミュレーションを行う．シミュレーションの結果として求まる工数，フォールト数が実際のプロジェクト PR の実績値に等しくなるようにパラメータの値を設定する．

パラメータ値の設定手順の概要は，次のとおりである．

- S1. 利用するプロジェクト PR とパラメータを定めるべき作業モデルを決める．
- S2. プロジェクト PR の実績データに基づいて，作業に携わる作業員の人数，割り当てられる作業量，入力プロダクトの完成度を決める．
- S3. 定めるべきパラメータの初期値を定める．
- S4. シミュレーションを100回行い，その結果として求まる工数，期間，残存フォールト数の値を収集する．
- S5. S4で得られた値とプロジェクト PR における実績の値を比較する．その差がある範囲内に入っていたら，設定を完了する．
- S6. 比較結果に基づいてパラメータの値を更新して，次は再び S4 へ．

この手続きによって，ある1つのプロジェクトのある作業員について，その作業を正確にシミュレートする発火レートのパラメータ値が求まる．

最終的には，個々のプロジェクトごとに設定されたパラメータの値から，同種のプロジェクトに対するパラメータの代表値を算出することも考えられる．

3.2 設定手順の詳細 (設計作業)

ここでは，設計作業を例としてパラメータ値の設定手順の詳細について述べる．設計作業で設定するパラメータは K_{cm} , K_{th} , K_{wr} , K_{in} であり，それぞれ，コミュニケーションレート，思考レート，記述レート，フォールトの混入率のパラメータを表している．

パラメータ K_{cm} , K_{th} , K_{wr} の値は工数や期間に影響を与える．例えば，設計作業においてコミュニケーションの発生頻度が高くなれば，時間が浪費される可能性も高くなると予想される．また，思考の発生頻度

が高くなれば、与えられた作業量が早く消費されるという解釈がなされる。つまり、 K_{cm} の値を増加させれば工数や期間は長くなる。一方、 K_{th} 、 K_{wr} の値が増加すれば工数や期間が短くなる。従って、これらの3つのパラメータの値は相互に依存するため同時に設定しなくてはならない。

今、次のような戦略 S が考えられる。

戦略 S: パラメータ K_{cm} 、 K_{th} 、 K_{wr} の内の2つについて値を固定し、残りの1つのパラメータの値を動かす。

この戦略 S の下で、設定手順 S4, S5, S6 を改訂する。

- S4. パラメータの値の各組合せに対し、工数の値に注目してシミュレーションを 100 回行う。
- S5. シミュレーションによる工数の値と PR での実績の値の差があらかじめ設定した範囲内に入ったときのパラメータの値を求める。
- S6. S5 で求めた組み合わせに対し (固定していた) パラメータの値を動かして、再び S4 へ。

しかし、この方法では時間がかかりすぎてしまう。そこで、3つのパラメータ K_{cm} 、 K_{th} 、 K_{wr} の工数、期間との関連性を考慮にいれて、計算時間を減らす次のような工夫をする。

IM1 K_{cm} 、 K_{th} 、 K_{wr} の中の1つだけその値を増加(または減少)させると、シミュレーションから求まる工数の値は単調に増加(または減少)することが分かっている。そこで、シミュレーションから求めた工数の値と実績値との差があらかじめ設定した範囲内に入れば、それ以降は調べる必要がない。

IM2 単純な方法で探すだけでは、いくつかの局所解が採用されてしまうことがある。こうした局所解を避ける工夫も必要となる。

- K_{cm} の値の増加に伴って、 r_{cm} は作業数に非常に敏感に反応するようになるし、作業結果も大きな影響を受ける。従って K_{cm} の値は他のパラメータ K_{th} 、 K_{wr} の値に比べて小さくする。また K_{cm} の値の決定に当たっては非常に小さな変動幅を用いる。
- K_{th} と K_{wr} の値を増加しても、 r_{th} と r_{wr} は作業数などにはあまり敏感に反応しないし、作業結果も余り大きな影響は受けない。

これらを考慮にいれて、戦略 S をさらに次のように具体化する。

設定手順 パラメータ K_{cm} の値を固定し、パラメータ K_{th} と K_{wr} を動かして調べて行く。なお、 K_{cm} の初期値を K_{th} や K_{wr} のものに比べて小さな値に設定しておく。まず求めた工数の値が PR での実績値に最も近いときの K_{th} と K_{wr} の値の組を探す。次に K_{cm} の値をごく小さい幅で動かして、上述の探索を繰り返す。最終的に求めた工数の値が実績値に最も近いときの K_{cm} 、 K_{th} 、 K_{wr} の値を解とする。

3.3 ツールの開発

前節で説明した設定手順に基づいて、パラメータの値を求めるツールを開発した。開発には、UNIX 上の Tcl/Tk を使用した。この理由は、パラメータの値を入力するときに GUI の使用が便利であること、シミュレータ自体がやはり Tcl/Tk を用いて作られているので、親和性が良くなること、などである。

4 提案法の適用実験

開発したパラメータ値計算のためのツールを用いて、実際のプロジェクトデータへの適用実験を行う。実験では、プロジェクトデータから人数、作業量などを抽出して設定手順 S1 から S6 を適用してパラメータ値を決定する。引き続いて、開発プロセスの設計部分(設計作業とコーディング)に対するシミュレーションをして、工数と残存フォールト数の評価を行う。

4.1 プロジェクト PR_1 、 PR_2

実験ではある企業から提供を受けた実際のソフトウェア開発プロジェクトのデータを使用する [6][12]。これらのプロジェクトは次に挙げるような特徴を持っている。

- プロジェクトは機器組込み型ソフトウェアの開発である。
- これらのプロジェクトはほぼ同種(同じ業務内容)のプロジェクトである。
- 同種のプロジェクトの間では、ほぼ同じ開発プロセス(図 2)が採用されている。

今回の実験では、2つのプロジェクト PR_1 、 PR_2 のデータを使用する。

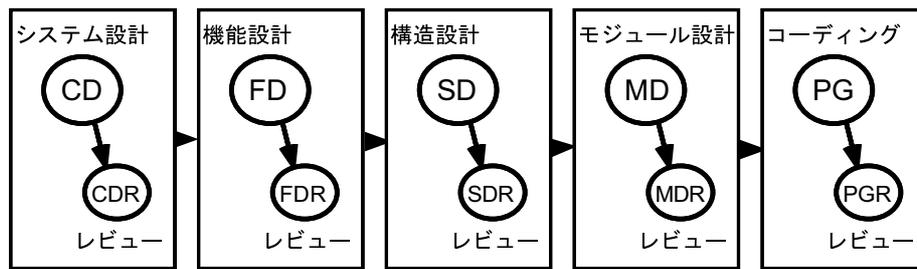


図 2: 設計工程のプロセス

4.2 パラメータの決定

開発したツールを利用して、プロジェクト PR_1 について各作業モデルの全てのパラメータ値の計算を行う。

ここではプロジェクト PR_1 のシステム設計 (CD) 作業を例にとって、パラメータ K_{cm} , K_{th} , K_{wr} , K_{in} を求めた様子を詳しく説明する。まず 3 つのパラメータ K_{cm} , K_{th} , K_{wr} が関連する工数を実測値に近付け、次にこれら 3 つのパラメータの値を利用して残存フォールト数に関するパラメータ K_{in} の値を設定する。

パラメータ K_{cm} , K_{th} , K_{wr} , K_{in} の値の設定に関して、次の P1-P6 の基本作業を行った。これらの作業の妥当性の根拠はプロジェクト PR_1 の実績値、これまでの経験 [8][10] などにある。

P1(作業特性の抽出) PR_1 のシステム設計 (CD) 作業の記録より、この作業は作業員 4 人によって行われていた。そして各作業員の作業時間の集計結果より、消費された作業量が 122 時間と分かる¹。また、作業にかかった工数は 17.4 人日であった。レビューとデバッグの結果から、この作業中に 23 個のフォールトが作り込まれていた。

P2(目標値と許容範囲の設定) P1 で示した実際のデータより、シミュレートされるシステム設計 (CD) 作業における工数の目標値を 17.4 人日とする。また目標値の $\pm 5\%$ 以内に入るようなパラメータの値を求めることにする。

P3(初期値の設定) これまでの使用経験よりパラメータ K_{cm} , K_{th} , K_{wr} の値はいずれも 0 より大きく 0.3 以下である。また、前にも述べたように、 K_{cm} は他の K_{th} , K_{wr} よりも小さな値が適している。これらより初期値を $(K_{cm}, K_{th}, K_{wr}) = (0.02, 0.05,$

$0.05)$ とする。また、 K_{in} は 1.0 以上の値をとるので、初期値を 1.0 とする。

P4(変動幅の設定) パラメータ K_{cm} , K_{th} , K_{wr} の値を少しずつ変動させて工数の値を目標値に近づけていく。この変動の幅は工数に与える影響を考慮して決める。 K_{cm} は大きな影響を及ぼすことが予想されるので、その変動幅は 0.02 刻みとする。それ以外については 0.05 刻みにする。なお、 K_{in} については、その値だけに依存してフォールト数が変化するので、2 分探索を用いる。

P5(最大値の設定) 初期値を決めたのと同様の理由でパラメータ K_{cm} , K_{th} , K_{wr} の最大値はすべて 0.30 とする。また、 K_{in} の最大値は 100 とする。

P6(工数に与える影響の分析) パラメータ K_{cm} の値を増加させると工数は増加するのに対し、パラメータ K_{th} , K_{wr} の値を増加させると工数は減少する。

戦略 S に基づいた設定手順をこれらの基本作業を踏まえて具体的に定め、パラメータ K_{cm} , K_{th} , K_{wr} , K_{in} の値を求めた。ここでは具体例を用いて設定手順の概要を説明する。実際にパラメータ K_{cm} , K_{th} , K_{wr} の値が求まる過程を表 1 に、 K_{in} の値が求まる過程を表 2 に示す。

まず、表 1 について説明する。初期値 $(K_{cm}, K_{th}, K_{wr}) = (0.02, 0.05, 0.05)$ で実行した時の工数は 103.26 となり、目標値の 17.4 からかけ離れている (表 1 の (1))。そこで、工数を減らすために K_{wr} の値を 0.05 刻みで増加させる。すると、工数の値は減少している。

K_{wr} を最大値の 0.30 に増加させても工数は 54.18 であり (表 1 の (2))、工数の目標値には遠い。そこで、今度は K_{th} の値を 0.05 だけ増加させて 0.10 とする。この状況 (つまり、 $K_{cm} = 0.02$, $K_{th} = 0.10$) で再び K_{wr} の値を 0.05 から 0.30 まで変動させる。

¹作業量はシミュレーションを実行する時に作業モデルに与える初期値となり、例えば 122 時間の作業量が消費された時に作業モデルの実行が終了する。

これを繰り返して行くと、 $(K_{cm}, K_{th}, K_{wr})=(0.02, 0.20, 0.30)$ の時に工数が 16.67 となる (表 1 の (3))。この時の値は工数の目標値の $\pm 5\%$ 以内という許容範囲に入っている。このパラメータ値の組は解の候補となり得るので保存する。表 1 中で太字で表した 5ヶ所がそうした解の候補である。

表 1: パラメータの計算過程

K_{cm}	K_{th}	K_{wr}	H(1) W1 = 17.4
0.02	0.05	0.05	103.26
0.02	0.05	0.10	71.26
0.02	0.05	0.15	62.97
0.02	0.05	0.20	57.36
0.02	0.05	0.25	56.24
0.02	0.05	0.30	54.18
0.02	0.10	0.05	64.79
0.02	0.10	0.10	42.92
0.02	0.10	0.15	35.18
0.02	0.10	0.20	32.57
0.02	0.10	0.25	29.81
0.02	0.10	0.30	29.13
0.02	0.15	0.05	53.80
0.02	0.15	0.10	32.37
0.02	0.15	0.15	26.46
0.02	0.15	0.20	23.58
0.02	0.15	0.25	22.19
0.02	0.15	0.30	21.20
0.02	0.20	0.05	48.70
0.02	0.20	0.10	29.50
0.02	0.20	0.15	22.98
0.02	0.20	0.20	20.09
0.02	0.20	0.25	18.34
0.02	0.20	0.30	16.67
0.02	0.25	0.05	45.59
0.02	0.25	0.10	26.19
0.02	0.25	0.15	20.30
0.02	0.25	0.20	17.41
0.02	0.25	0.25	15.70
0.02	0.30	0.05	42.71
0.02	0.30	0.10	25.01
0.02	0.30	0.15	19.01
0.02	0.30	0.20	15.75
0.04	0.20	0.30	19.26
0.04	0.25	0.25	17.92
0.04	0.25	0.30	16.82
0.04	0.30	0.20	18.20
0.04	0.30	0.25	16.36
0.06	0.25	0.30	19.01
0.06	0.30	0.25	19.17
0.06	0.30	0.30	16.42
0.08	0.25	0.30	19.83

この実験では試行の回数を減らすために幾つかの工夫をしている。例えば、表 1 の (5) では、 $K_{wr} = 0.25$ の時に工数が 15.7 となっている。これ以後は K_{wr} の値を増加させても工数は減るだけなので $K_{wr} = 0.30$ とする組についてのシミュレーションを省略している。また、表 1 の (6) で $K_{cm} = 0.04$ と増加させた後、それまでに工数の目標値を上回ってしまった K_{th} と K_{wr} の値の組についても全て省略している。

この結果、工数の目標値の $\pm 5\%$ 以内に入った 5 個のパラメータ値の組で最も目標値に近いのは $(K_{cm}, K_{th}, K_{wr})=(0.02, 0.25, 0.20)$ のときである (表 1 の (4))。そこでこの値を PR_1 のシステム設計 (CD) 作業におけるパラメータの設定値として採用する。

次に、表 2 について説明する。表 1 の計算で得られたパラメータ K_{cm}, K_{th}, K_{wr} の値を用いて、パラメータ K_{in} の値の設定を行う。 K_{in} によるフォールト数の目標値は 23 として、2 分探索法を用いて K_{in} の値を求める。探索を終了させるのは、フォールト数が目標値の 5% 以内に入った時とする。表 2 に示す計算を経て、最終的に $K_{in}=18.75$ が得られた。

なお、表 2 には参考のためにその時点での工数の値も記してある。初めに考察した通り、工数の値は K_{in} の影響をほとんど受けていない。

表 2: K_{in} の計算過程

K_{in}	ctH[g] W1 = 23	H(Q1)
1.00	0.00	17.63
100.00	130.23	17.43
50.00	63.76	17.77
25.00	30.26	17.70
12.50	15.52	17.94
18.75	23.75	17.80

こうしてプロジェクト PR_1 の各作業 CD, FD, SD, MD, PG ごとに得られたパラメータの値を表 3, 4 に示す。

4.3 実験結果の考察

この実験を行って新たに明らかになったことを以下にまとめる。

表 3: 各設計作業毎のパラメータ

	K_{cm}	K_{th}	K_{wr}	K_{in}
CD	0.02	0.25	0.20	18.75
FD	0.04	0.30	0.25	12.50
SD	0.04	0.30	0.20	9.38
MD	0.10	0.20	0.20	33.75
平均値	0.05	0.263	0.215	18.6

表 4: コーディング作業のパラメータ

	K_{cm}	K_{th}	K_{wr}	K_{in}
$Coding(PG)$	0.04	0.20	0.15	43.75

(1) 工数への影響

K_{cm} が増加した時に工数が増加する割合が大きいことに考慮した調整をしてきた。従って、 K_{th} と K_{wr} の影響については、単にこれらの値が増加すれば工数の減少が起きる、というものであった。しかし、実際には K_{th} と K_{wr} ではその影響の大きさに著しい差が見られた。つまり、 K_{th} を増加させた時の工数の減少の割合は K_{wr} を増加させた時のそれよりも相当に大きくなっていった。

(2) フォールト数への影響

フォールト数は、パラメータ K_{in} の値だけでなく、パラメータ K_{wr} の値からも影響を受けることが分かった。その原因は、作業モデル上では記述作業に対応するトランジションが発火した時にフォールトの混入も確率的に起こるとしているためである。従って、総フォールト数はフォールトの混入率 p_{in} だけでなく、発火レート r_{wr} にも関わることに留意したパラメータ値の設定手順の開発が必要である。

5 求めたパラメータ値の評価

5.1 評価実験

前節で得られたパラメータの値を用いて、プロジェクト全体（つまり、図 2 に示すプロセス全体）のシミュレーション実験を行う。ここでは以下を確認することを実験の目的とする。

- 個々の作業モデル毎に（つまり、局所的に）設定したパラメータ値を利用してプロセス全体を作り上げた時にうまく動かか。つまり、最終プロダクトの品質とプロジェクトの工数の総和で評価しても満足できるシミュレーション結果を保証できるのか。

ここでは、プロジェクト PR_1 から得られたパラメータ値をプロジェクト PR_2 に適用してコーディング作業終了時までのプロセス（図 2 の CD , CDR , FD , FDR , SD , SDR , MD , MDR , PG , PGR の部分）をシミュレートして、その時点での工数と残存フォールト数を実績値と比較する。なお、ここで適用するパラメータの値は PR_1 の個々の設計作業毎に得られたパラメータ値を平均したもの（表 3）を PR_2 の各設計作業に与え、コーディング作業については表 4 のものをそのまま PR_2 に適用する。

なお、今回のパラメータ値の設定では、レビュー作業は議論の対象としなかった。しかし、シミュレーションにおいてレビューを外すことができないため、レビュー作業のパラメータは従来のもの [7][8][9][10] を仮に使用することにした。

5.2 分析

表 5 にプロジェクト PR_2 について行ったシミュレーション結果を示す。ここで、残存フォールト数とは、設計作業が終了した時点でプロダクト中に含まれていたと考えられるフォールト数のことである。その実績値は以降のデバッグ工程と受入テストにおいて発見された数の合計を示している。

表 5: シミュレーション結果

	工数 (人日)	残存フォールト数
実績値	41.7	17
シミュレーション	38.6	17.9

表 5 にまとめたシミュレーション結果について考察する。結果からは工数と残存フォールト数のどちらについても非常に実績値に近い値を得ることができたと言える。このことから、提案する方法による設計工程とコーディング工程のパラメータの決定がソフトウェアプロダクトの品質とコストの予測に有効であることが確認できた。

6 まとめ

本研究では、ソフトウェア開発プロジェクトシミュレータのためのパラメータ値の設定手順を提案した。提案した手順で求めたパラメータ値を用いた適用実験を行い、この手法の有効性についても確認した。

今後の課題としては以下のことが挙げられる。

- (1) レビュー作業モデルに含まれる発火レートパラメータの値の設定法について検討すること。
- (2) 今回の実験中に判明したパラメータの特性を考慮して提案方法を改良すること(特に、フォールト数への影響については早急に対処する必要がある)。
- (3) 設計工程以外の工程にも提案法を適用して最適なパラメータの値を求めてみること。

謝辞

本研究において利用しているデータの提供に関して多大のご協力を頂いたオムロン株式会社 坂本啓司氏、高木徳生氏に深謝致します。

参考文献

- [1] 青山幹雄, 内平直志, 平石邦彦: “ペトリネットの理論と実践,” 朝倉書店 (1995).
- [2] L. Briand, K. E. Eman, O. Laitenberger and T. Fussbroich: “Using simulation to build inspection efficiency benchmarks for development projects,” Proc. of the 20th ICSE, pp.340–349 (1998).
- [3] F. P. Brooks Jr.: “The Mythical Man-Month,” Addison Wesley (1975).
- [4] T. Furuyama, Y. Arai and K. Iio: “Fault generation model and mental stress effect analysis,” The Journal of Systems and Software, vol.26, pp.31–42 (1994).
- [5] N. Hanakawa, S. Morisaki and K. Matsumoto: “A learning curve based simulation model for software development,” Proc. of the 20th ICSE, pp.350–359 (1998).
- [6] 岸田孝一監修 (アルフォンゾ・フェジッタ, アレキサンダー・ウルフ編): “ソフトウェアプロセスのトレンド,” 海文堂 (1997).
- [7] S. Kusumoto, O. Mizuno, Y. Hirayama, T. Kikuno, Y. Takagi and K. Sakamoto: “A new project simulator based on generalized stochastic Petri-net,” Proc. of the 19th ICSE, pp.293–303 (1997).
- [8] 水野修, 平山裕司, 楠本真二, 菊野亨, 高木徳生, 坂本啓司: “ソフトウェア開発プロセスの進捗予測システムの開発,” ソフトウェアシンポジウム'96 予稿集, pp.90–98 (1996).
- [9] 水野修, 楠本真二, 菊野亨: “ソフトウェア開発プロセスの並行実行に対する拡張一般化確率ネットに基づく評価,” 信学技報 SS97-5, pp.33–42 (1997).
- [10] O. Mizuno, S. Kusumoto, T. Kikuno, Y. Takagi and K. Sakamoto: “Estimating the number of faults using simulator based on generalized stochastic Petri-net,” Proc. of the Sixth Asian Test Symposium, pp.269–274 (1997).
- [11] H. Sackman, W. J. Erickson and E. E. Grant: “Exploratory experimental studies comparing online and offline programming performance,” Commun. ACM, vol.11, no.1, pp.3–11 (1968).
- [12] T. Tanaka, K. Sakamoto, S. Kusumoto and T. Kikuno: “Improvement of software process by process visualization and benefit estimation,” Proc. of the 17th ICSE, pp.123–132 (1995).