社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
KBSE98-34 (1998-11)

学習習熟を考慮したソフトウエア開発シミュレーションモデルの評価

花川典子　松本健一　鳥居宏次
奈良先端科学技術大学院大学情報科学研究科
〒630-0101 奈良県生駒市高山町 8916-5
E-mail{noriko-h,matumoto,torii}@is.aist-nara.ac.jp

あらまし　本稿では,筆者らが既に提案している学習習熟を考慮したソフトウエア開発シミュレーションモデルの評価を行う. 具体的には次の２つの段階を経て, プロジェクト管理者による開発作業時間見積り値との適合度を検定する. 第１段階では, プロジェクト管理者がモデルのパラメータ値を容易に設定できるパラメータ変換式を定める.第２段階では, パラメータ変換式で求めたパラメータ値を使ったシミュレーション結果の開発作業時間見積り値とプロジェクト管理者による見積り値との適合度を検定する.評価の結果,変換式を含む提案モデルとプロジェクト管理者の６件の見積り値は有意水準5%で適合することが確認できた.

キーワード　シミュレーションモデル, 学習習熟, モデル評価, パラメータ変換式, バーチャルプロジェクト

Evaluation of a Learning Curve Based Simulation Model for Software Development

Noriko Hanakawa Ken-ichi Matumoto Koji Torii
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0101, Japan
E-mail{noriko-h,matumoto,torii}@is.aist-nara.ac.jp

Abstract　This paper evaluates a software development simulation model which is based on developer's learning curve. Concretely, via the following two steps, goodness of fit of the proposed model is clarified. In the first step, conversion formulas which can calculate values of the parameters in the model using questionnaires to managers are defined. In the second step, the goodness of fit of the model is tested using the values which are calculated in the conversion formulas. The results show that delivery time estimated by the model including the conversion formulas is fitted in with ones estimated by six project managers.

key words　Simulation model , Learning curve, Evaluation of model, conversion formulas, Virtual project

1. はじめに

近年,「良いものを早くほしい」という顧客の要望を満たすために,ソフトウエア開発技術者はコンポーネントウエアやCORBAなど新技術を駆使し,高品質,高生産性を達成しなくてはならない.そのため,開発者たちはプロジェクトにおいてソフトウエア開発と同時に新技術の習得も要求されている[1][8].

新技術習得が必要なプロジェクトの開発者の生産性は新技術習熟の度合いによって変化する.プロジェクト初期では新技術の獲得や知識の不足による誤りで時間が費やされ生産性が低い.しかし、プロジェクト後半では新技術獲得に費やされる時間は減少し,また知識不足による誤りも少なくなり結果的に生産性が向上する.

そこで筆者らは,開発者の学習習熟を考慮したソフトウエア開発シミュレーションモデルを提案した[2].提案モデルでは,開発者の作業に対する知識と作業実行に要求される知識の関係から,生産性と習熟の度合いを求める.つまり,開発者が難しい作業を実施する時,生産性は低いが開発者の知識は増加して習熟し,反対に開発者にとって容易な作業を実施する時,生産性は高いが開発者の知識は増加せず習熟もしないという状況をモデル化した.提案モデルで様々なプロジェクトを想定してシミュレーションを行ったが,シミュレーション結果の見積り値と実際のプロジェクト管理者の見積り値を比較して評価することができなかった.なぜならば,プロジェクト管理者にとってモデルのパラメータ値設定が非常に困難であったからだ.

本稿では,筆者らが既に提案している学習習熟を考慮したソフトウエア開発シミュレーションモデルの評価を行う.具体的にはモデルによる開発作業時間見積り値とプロジェクト管理者による開発作業時間見積り値との比較を行う.評価の手順は2段階に分かれる.

評価の第1段階ではプロジェクト管理者が容易にパラメータ値を求めることができるパラメータ変換式を求める.変換式は,Web上に設定された作業の内容や開発者の経験年数などの特徴が示されたバーチャルプロジェクトに対するプロジェクト管理者への簡単な質問からモデルのパラメータ値を導く.パラメータ変換式はモデルのパラメータ毎に固有の補正係数を持ち,補正係数はモデルのシミュレーション結果の見積り値とプロジェクト管理者の見積り値との差が最小になるように設定される.

評価の第2段階ではパラメータ変換式を含めた学習習熟を考慮するソフトウエア開発シミュレーションモデルを評価する.バーチャルプロジェクトに対して,パラメータ変換式から求めたパラメータ値でモデルをシミュレーションした結果の開発作業時間見積りと,第1段階とは別のプロジェクト管理者による開発作業時間見積りとの適合度を検定し,パラメータ変換式を含む提案モデルを評価する.

以下,2章ではすでに提案したソフトウエア開発シミュレーションモデルの概要を説明し,3章では評価の手順を示す.4章で評価手順に従ったパラメータ変換式を含む提案モデルの評価を行い,5章でまとめる.

2. 提案モデル

開発者の学習習熟を考慮した提案モデルでは,プロジェクト実施中の開発者の習熟や生産性の変化を明らかにし,プロジェクトの進捗の様子を示す.提案モデルは,作業の難しさとその作業量を示す作業モデルと,開発者の知識獲得効率を示す知識モデルと,作業の難しさによって変化する開発者の生産性を表す生産性モデルから構成される.

2.1. 作業モデル

作業モデルは作業実施に要求される知識レベル,すなわち難しさと,その作業量の関係を示す.例えば,Cプログラム開発の場合多くの関数を作成するが,それぞれの関数の難しさは異なる.非常に簡単な宣言だけの関数もあれば,複雑な処理の関数もある.非常に容易もしくは困難な関数が少なく,中程度ほどの関数が多いと仮定すると,作業モデルは関数の難しさの分布が正規分布の式となる.(図1参照)

$$W_j(\theta) = w_j \frac{1}{\sqrt{2\pi\sigma}} e^{\left[-\frac{(\theta-\mu)^2}{2\sigma^2}\right]}$$

θ:作業実行に要求される知識レベル

$W_j(\theta)$:要求される知識レベルがθの作業量

w_j:総作業量

μ:知識レベルの平均

σ:知識レベルの分散

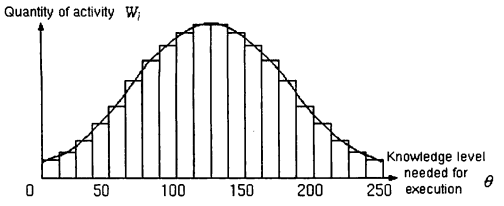

図 1 作業モデル

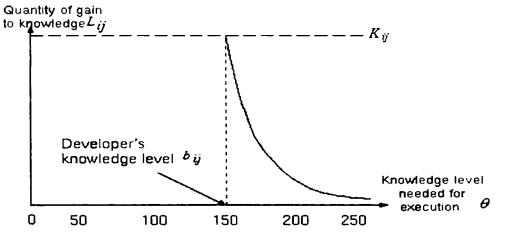

図2 知識モデル

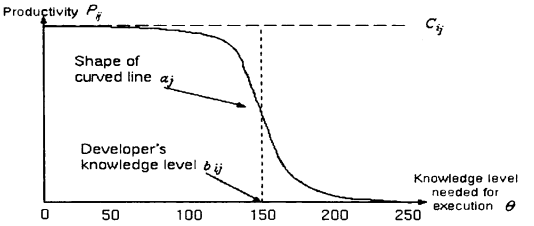

図 3 生産性モデル

2.2. 知識モデル

知識モデルは作業実施による開発者の知識獲得の様子を示す. 例えば,容易な作業を行う場合,開発者は新しく学習する内容はなく知識は増えない. 反対に難しい作業を行う場合,開発者は未知の知識を要求され学習により知識が増加する. ただし,開発者にとって難しすぎる作業の場合は理解困難となり知識獲得量は少ない. 開発者の知識獲得の様子を表す知識モデルを以下に示す（図2参照）.

$$L_{ij}(\theta) = w_j \begin{cases} K_{ij} \times e^{-E_{ij}(\theta - b_{ij})} & (b_{ij} \le \theta) \\ 0 & (b_{ij} > \theta) \end{cases}$$

θ:作業実行に要求される知識レベル

$L_{ij}(\theta)$:要求される知識レベルが θ の知識獲得量

b_{ij}:開発者の知識レベル

E_{ij}:開発者の知識獲得効率

K_{ij}:開発者の単位時間あたりの最大知識獲得量

2.3. 生産性モデル

生産性モデルは開発者の知識レベルと作業実行に要求される知識レベルの関係から開発者の生産性の変化を示す. 例えば,開発者にとって容易な作業を行う場合は生産性は高く,難しい作業を行う場合は生産性は低い. 前述した知識モデルの知識獲得量と反対になっている. 生産性モデルをオージブモデル（累積正規分布）[7]を用いて次のように示す（図3参照）.

$$P_{ij}(\theta) = C_{ij} \int_{-\infty}^{a_j(b_{ij} - \theta)} \frac{1}{\sqrt{2\pi}} e^{\frac{-t^2}{2}} dt$$

θ:作業実行に要求される知識レベル

$P_{ij}(\theta)$:要求される知識レベルが θ の生産性

C_{ij}:開発者の最大の生産性

a_j:作業実行に要求される作業の厳密さ

b_{ij}:開発者の知識レベル

2.4. シミュレーション手順

作業モデル,知識モデル,及び生産性モデルを用いて以下の手順でシミュレーションし,作業の進捗状況や開発者の習熟の様子を明らかにする.

手順１：各モデルの初期化と時間 t の初期化

手順２：時間 t に実施する作業の選択（θの決定）

手順３：$P_{ij}(\theta)$の決定

手順４：$L_{ij}(\theta)$の決定

手順５：開発者の知識レベルb_{ij}の更新

$b_{ij}=b_{ij}+L_{ij}(\theta)$

手順６：作業モデルの更新

$W_j(\theta)=W_j(\theta)-P_{ij}(\theta)$

手順７：知識モデルと生産性モデルを更新

手順８：終了判定

2.5. 提案モデルの関連研究

ソフトウエア開発プロセスのモデル化と評価に対しいくつかの方法が提案されている. Kellnerは,機能,動作,構成の３つの側面から開発プロセスをモデル化し,そのうち主に動作面のモデルに基づいてプロセスシミュレーションを行っている[5]. シミュレーションには,通信ソフトウェア設計用のCASEツールの一つであるSTATEMATEが用いられ,開発工数の見積もりや作業計画の作成が可能となっている. STATEMATEは状態遷移モデルに基づくツールである. 従って,状態遷移確率等によって作業効率や作業者の習熟度を表現することは原理的に可能であるが,特に言及はされていない.

Kusumotoらは,一般化確率ペトリネットで作業進捗をモデル化している[6]. ソフトウェア開発を構成する各作業に資源(作業者等)を割り当てた上でペトリネットの発火機構を利用すると,開発

期間,開発工数,及び,作成されるソフトウェアの品質を推定することが可能である．パラメータとして「作業者の経験レベル」がモデルに組み込まれている．「作業者の経験レベル」は,ペトリネットの発火レートや作成されるソフトウェアへのフォールトの混入率を決定する重要なパラメータの一つとされている．但し,その値は作業者毎に与えられる定数であり,シミュレーションにおいて変化することはない．

Iidaらは,多人数での並行作業による開発を記述するためのプロセスモデル,及び,並行作業間の影響を記述するための作業進捗モデルを提案している[3]．彼らのモデルを用いれば,並行作業による開発時の期間と工数の関係を予測することが可能である．作業者の習熟度を表すパラメータとして「スタッフの能力」がモデルに組み込まれている．「スタッフの能力」は,作業効率(単位時間当りの作業進捗)を決定する重要なパラメータの一つとされている．但し,Kusumotoらのモデル同様,その値は作業者毎に与えられる定数であり,シミュレーションにおいて変化することはない．

3. 評価手順

3.1. 概要

バーチャルプロジェクトに対する提案モデルのシミュレーションによる開発作業時間見積りと,見積もりのエキスパートであるプロジェクト管理者による開発作業時間見積りの２つの見積り値を比較して,提案モデルの適合度を検定し評価する．

そのため,提案モデルへバーチャルプロジェクトに関するパラメータ値を与える必要があるが,プロジェクト情報から直接パラメータ値を求める方法が提案モデルにはない．そこで,見積り経験豊かなエキスパートのプロジェクト管理者がバーチャルプロジェクトの情報をもとにパラメータ値を設定する．この方法で設定されたパラメータ値が最もバーチャルプロジェクトの特徴を反映できると思われる．

しかし,提案モデルのパラメータである単位時間当たりの最大知識獲得量（K_{ij}）や知識獲得効率（E_{ij}）などの値を求めることはモデルを十分理解する必要があり,プロジェクト管理者にとって非常に困難である．そこで,プロジェクト管理者がパラメータ値を容易に設定できるパラメータ変換式を作成する．この変換式はプロジェクト管理者のパラメータに関する簡単な質問の回答を得

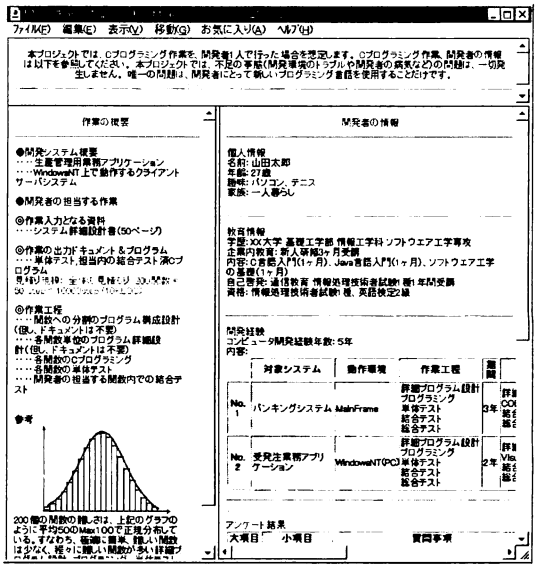

図４ バーチャルプロジェクト画面

るだけでパラメータ値を求めることができる．パラメータ変換式から求められたパラメータ値で提案モデルのシミュレーションを行い開発作業時間見積り値を求め，プロジェクト管理者の開発作業時間見積り値と比較して提案モデルを評価する．

評価はWeb上に設定されたバーチャルプロジェクトに基づき行われる．また，評価の手順は提案モデルのパラメータ変換式を求める第１段階と,パラメータ変換式を使用して提案モデルを評価する第2段階に分かれる．まず,バーチャルプロジェクトを説明し,評価手順を示す．

3.2. バーチャルプロジェクト

Web上のバーチャルプロジェクトは見積りの対象となるプロジェクトの内容を詳細に示す．図４のバーチャルプロジェクト画面では,画面上部にプロジェクト全体の特徴を示す．例えば設定されたプロジェクトは５人の開発者が１つの作業を行うプロジェクトであり,突発的な問題（ハードウエアトラブル）は発生しないこと等を明記する．

画面左下側は作業に関する情報が示される．たとえば,詳細設計とプログラミングとテストという作業の種類や作業の規模,プログラミング言語など使用する開発技術が明記する．

画面右下には開発者に関する情報が示される．開発者の名前や家族構成などの基本的な情報,ソフトウエア開発経験年数や経験したプロジェクトの種類や開発規模実績などの値や,更に開発者本

人へのアンケート結果として性格や興味を持つ開発技術,今後の作業への希望も示す.

本来ならば,過去に実施されたプロジェクトの作業時間実績と提案モデルでの見積り時間を比較するべきである.しかし,提案モデルには具体的なプロジェクト情報からパラメータ値を導き出す方法がないので,複雑な環境下で実施された現実のプロジェクト実績を直接利用するのは難しい.単純で明確な特徴を示すバーチャルプロジェクトで評価した後,過去に実施されたプロジェクトの開発作業時間への適用を行う予定である.

3.3. 評価の第一段階:パラメータ変換式

第1段階ではプロジェクト管理者への簡単な質問の回答よりモデルのパラメータ値を導くパラメータ変換式を求める.プロジェクト管理者は変換式を通しモデルにパラメータ値を容易に与えられる.変換式はプロジェクト管理者への質問内容から導かれる基本型変換式と補正係数を使って調整した拡張型変換式に別れる.それぞれを説明する.

(1)基本型パラメータ変換式

モデルのパラメータの意味を実際のプロジェクトでも推測可能な質問内容に変えたことで基本型変換式が必要となる.提案モデルの質問内容とパラメータ基本型変換式の例を以下にに示す.

・C_{ij}(最大の生産性)

質問内容:開発者の最高生産性は?

基本型変換式:C_{ij}=入力値

・K_{ij}(単位時間あたりの最大知識獲得量)

質問内容:新技術を完全に習得するために必要な日数は?(習得中は作業しない)

基本型変換式:K_{ij}=100/(入力値×8)

基本型変換式はC_{ij}のように,プロジェクト管理者がよく使う値であるならば,容易にその値が推測できるので直接その値を質問する.従って基本型変換式=入力値である.K_{ij}パラメータのようにプロジェクト管理者に馴染みのない内容は質問内容をわかりやすくする必要がある.K_{ij}(単位時間あたりの最大知識獲得量)では,新技術を完全に習得するために必要な日数(習得中は作業しない)を問う.つまり,作業を行わないでの毎日最大限の知識獲得ができると考え,知識が100%になるまでの時間を質問する.従って1時間あたりの最大知識獲得量は,100%を(入力日数×8時間)で割ることとなり,上記の基本型変換式となる.

(2)拡張型パラメータ変換式

基本型パラメータ変換式を元に,補正係数を使って調整した拡張型パラメータ変換式を求める.前項ではプロジェクト管理者に理解しやすい質問を作成したが,それでも誤った解釈による偏りがおきる.そこで入力された値には誤差があると仮定して,基本型パラメータ変換式に補正係数を付加する.付加は様々な方法があるが,ここでは単純に乗算とする.

拡張型パラメータ変換式=

基本型パラメータ変換式×補正係数

次に補正係数の値を求める.バーチャルプロジェクトに対するプロジェクト管理者の見積り値は正しく,かつプロジェクト管理者は基本型変換式に基づく質問に対し一定の偏りを持って回答すると仮定する.補正係数はモデルのパラメータの数だけあり,それぞれの補正係数を様々に変化させてシミュレーションした結果の見積りと,プロジェクト管理者の見積りとの差が最小になる補正係数の組み合わせが,求める補正係数の値である.従って,次の式が最小になる補正係数値の組み合わせを求める.

$$1/k \times \Sigma^{k}_{j=1} |(S_{jn} - P_j)| / P_j$$

k:バーチャルプロジェクト数

S_{jn}:補正係数組み合わせパターンnのプロジェクトjのモデルのシミュレーション結果

P_j:プロジェクトjのエキスパートの見積り

3.4. 評価の第二段階:提案モデルの評価

第一段階で求めた拡張型パラメータ変換式を含む提案モデルと,第1段階で登場しないプロジェクト管理者の見積り値との適合度を検定し,提案モデルを評価する.

拡張型パラメータ変換式を求めなかったプロジェクト管理者がバーチャルプロジェクトを見積る.更にこの管理者がパラメータ変換式の元になるパラメータに関する質問に回答する.回答された値を拡張型パラメータ変換式でモデルのパラメータ値に変換し,提案モデルのシミュレーションでバーチャルプロジェクトの見積り値を得る.両者の見積り値の適合度を検定で求める.

4. 評価実験

前述した評価手順で提案モデルを評価する.

4.1. 評価の第一段階:パラメータ変換式

(1)基本型パラメータ変換式

提案モデルのパラメータに関する質問と基本型パラメータ変換式を以下に示す.作業に関する4

表２ 補正係数を変化させた時のモデル計算結果と見積り値との差

補正係数組合ﾊﾟﾀｰﾝ	補正係数W	補正係数B	補正係数C	補正係数K	補正係数E	モデルのｼﾐｭﾚｰｼｮﾝ結果(月)		ｴｷｽﾊﾟｰﾄ見積りとの差
						VP_1	VP_2	
ﾊﾟﾀｰﾝ88	15	1	1	2.5	1.00	6.26	1.61	0.2069
ﾊﾟﾀｰﾝ89	15	1	1	2.5	1.25	6.57	1.81	0.1432
ﾊﾟﾀｰﾝ90	15	1	1	2.5	1.50	2.46	6.69	0.0079
ﾊﾟﾀｰﾝ91	15	1	1	2.5	1.75	2.51	7.15	0.0389
ﾊﾟﾀｰﾝ92	15	1	1	2.5	2.00	2.90	7.22	0.1235

つのパラメータ(作業モデルのw_j, μ, σと生産性モデルのa_j)も質問対象にする必要があるが,実験を簡単にするため開発者に関するパラメータに限定する.

・生産性モデルのC_{ij}（最大の生産性）

質問内容：開発者の最高生産性は？

変換式：C_{ij}=入力値

・知識モデルと生産性モデルb_{ij}（知識レベル）

質問内容:現在の開発者の新技術知識は何%?

変換式：b_{ij}=入力値

・K_{ij}（単位時間あたりの最大知識獲得量）

質問内容：新技術を完全に習得するために必要な日数は？(習得中は作業しない)

変換式：K_{ij}=100/(入力値×8)

・E_{ij}（知識獲得効率）

質問内容：どのくらい難しい作業を要求されると全く作業できない状態になるか？ 作業実行に要求される知識レベル%と開発者の知識レベル%の差を答えて下さい.

（知識モデルでL_{ij}≦0.01時の開発者と作業の知識の差を入力させる）

変換式:E_{ij}=2/(入力値)$\mathrm{Log}_e 10$

(2)拡張型パラメータ変換式

見積り経験5年以上のエキスパートである3人プロジェクト管理者が２つのバーチャルプロジェクトの開発作業時間を見積る.各作業時間見積りを表１に示す. バーチャルプロジェクト１(VP_1)の内容は,開発経験5年でCOBOLとVisualBasicで詳細設計,プログラミング,テストの経験がある開発者が,初めてC言語による詳細設計,プログラミング,単体テスト作業を行う. C言語の知識は企業内新人研修を1ヶ月ほど受けただけである. 開発者の担当する見積り規模はおよそ10KLOCである. バーチャルプロジェクト２(VP_2)は見積り規模が1KLOC以外は同様の設定である.

次にエキスパートに対して開発者に関する質問を行う. VP_1におけるエキスパートによるC_{ij}入力値,b_{ij}入力値,K_{ij}入力値,及びE_{ij}入力値を表１に示す. 更に,拡張パラメータ変換式の補正係数を求める. 各パラメータの拡張型パラメータ変換式を以下に示す.

表１ エキスパートのプロジェクトのパラメータ入力値と見積もり値

ﾌﾟﾛｼﾞｪｸﾄ管理者	C_{ij}入力値	b_{ij}入力値	K_{ij}入力値	E_{ij}入力値	VP_1見積り	VP_2見積り
A	1.5	10	25	30	7.5	2.5
B	2.0	10	18	20	5.8	2.0
C	2.5	8	60	40	6.7	3.0

拡張型変換式: w_j=プログラム行数×補正係数W

拡張型変換式:b_{ij}=入力値×補正係数B

拡張型変換式:C_{ij}=入力値×補正係数C

拡張型変換式:K_{ij}=100/(入力値×8)×補正係数K

拡張型変換式:E_{ij}=2/(入力値)$\mathrm{Log}_e 10$×補正係数E

それぞれの補正係数を独立で変化させて,補正係数の組合せﾊﾟﾀｰﾝnを作る. 組合せﾊﾟﾀｰﾝの数を以下の式に示す.

組合せﾊﾟﾀｰﾝ数=(Wの変化回数)×(Bの変化回数)×(Cの変化回数)×(Kの変化回数)×(Eの変化回数)

入力値から拡張型パラメータ変換式求めたパラメータ値で提案モデルを組合せﾊﾟﾀｰﾝ数の回数シミュレーションする. シミュレーション結果の見積り値とエキスパートが見積った値との差を表２のエキスパート見積り値との差の項に示す.

各補正係数の変化させてシミュレーションすると,エキスパート見積り値との差が最も小さい補正係数の組み合わせはパターン90のW=15, B=1, C=1, K=2.5, E=1.5となる.その時のエキスパート見積り値との差は0.0079となり,シミュレーション結果と見積もりの誤差は約0.8%である. この補正係数を採用すると拡張型パラメータ変換式は以下のとおりになる.

変換式:w_j=プログラム行数×15

変換式:C_{ij}=入力値×1

表3 プロジェクト管理者の見積りとモデルのシミュレーション結果

管理者	ﾌﾟﾛｼﾞｪｸﾄ	C_{ij}入力値	b_{ij}入力値	K_{ij}入力値	E_{ij}入力値	見積り値 (月)	モデル結果(月)
E	VP_1	2.5	10	83	30	10.0	8.2381
F	VP_1	0.95	20	25	60	12.5	11.144
G	VP_1	3.0	10	83	30	8.0	7.4503
H	VP_1	2.0	20	90	20	10.0	10.342
I	VP_2	0.95	20	25	60	2.5	1.6059
J	VP_2	3.0	10	83	300	4.0	4.3533

変換式:b_{ij}=入力値×1

変換式:K_{ij}=100/(入力値×8)×2.5

変換式:E_{ij}=2/(入力値)Log_e10×1.5

4.2. 評価の第二段階：提案モデルの評価

拡張型パラメータ変換式を含む提案モデルを,他の6人のプロジェクト管理者のバーチャルプロジェクトに対する開発作業時間見積りで検証する.

6人のプロジェクト管理者の見積りと4つの質問に対する入力値,およびモデルでのシミュレーションによる見積り値を表3に示す. この6件の見積りに関して,拡張型パラメータ変換式を含む提案モデルの適合度の検定を行う.

プロジェクト管理者の見積り値が変換式を含むモデルと適合しないという仮説をたてる. ピアソン型検定[4]の統計量を下記の式で求める.

$$\chi^2=\Sigma^{k}_{j=1}((n_j-np_j)^2/\ np_j)$$
$$=(10.0-8.24)^2/8.24+(12.5-11.1)^2/11.1+$$
$$(8.00-7.45)^2/7.45+(10.0-10.3)^2/10.3+$$
$$(2.50-1.61)^2/1.61+(4.00-4.35)^2/4.35$$
$$=1.13$$

k:測定回数

n_j: j回目の測定値

np_j: j回目の理論値

計算の結果,統計量χ^2=1.13となり,自由度5の時の有意水準5%でのカイ二乗分布$\chi^2_{0.05}$=1.15よりも小さいので有意である. 従って,見積り値が変換式を含むモデルと適合しないという仮説は棄却され,6件のプロジェクト管理者の見積り値がパラメータ変換式を含むモデルと適合すると評価できる.

5. まとめ

本稿では, 筆者らが既に提案している学習習熟を考慮したソフトウエア開発シミュレーションモデルのパラメータ変換式を求めた. そして,バーチャルプロジェクトのプロジェクト管理者による6件の見積り値と提案モデルとの適合度をピアソン型検定で検証した. その結果,6件の見積り値は提案モデルに適合することがわかった.

今後はWeb上のバーチャルプロジェクトを利用して作業に関するパラメータ変換式や他の種類の作業や知識に関するパラメータ変換式を求める. 更に,本評価をWeb上で公開して(http://noripc1.aist-nara.ac.jp/HyoukaSystem/LCBModel/index.html)モデルの試行を可能にし,経験豊富なプロジェクト管理者からの意見をモデルに反映させる予定である.

文 献

[1]B. Bochenski, "Implementing Production-quality Client/Server Systems," John Wiley & Sons,1994.

[2] N. Hanakawa, S. Morisaki, K. Matumoto, "A Learning Curve Based Simulation Model for Software Development," Proceedings of 20th International Conference on Software Engineering,pp.350-359,1998.

[3] H. Iida, J. Eijima, S. Yabe , K. Matsumoto, K. Torii, " Simulation model of overlapping development process based on progress of activities," Proceedings of 1996 Asia-Pacific Software Engineering Conference,pp.131-138,1996.

[4] 池田央, "統計ガイドブック",新曜社,1989

[5] M. Kellner, "Software process modeling support for management planning and control," Proceedings of 1st International Conference on Software Process,pp.8-28,1991.

[6] S. Kusumoto, O. Mizuno, T. Kikuno, Y. Hirayama, Y. Takagi, K. Sakamoto, "A new software project simulator based on generalized stochastic," Proceedings of 19th International Conference on Software Engineering, pp.293-302,1997.

[7]芝祐順, 渡部洋, 石塚智一, "統計用語辞典", 新曜社,1980.

[8]E. Yourdon, " Object-Oriented Systems Design," Prentice Hall,1994.