1D－5

作業の並行化による影響を考慮した開発プロセスシミュレータの実験的評価

矢部 智†　飯田 元‡　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
‡奈良先端科学技術大学院大学 情報科学センター

1 はじめに

ソフトウェア開発においては，上流工程の作業の完了を待たずに下流工程の作業を開始すること (作業の並行化) によって，開発期間の短縮化が図られることが多くなってきている [1]．しかし，この方法は開発総工数の増加を招くため，実際のプロジェクトにおいては，開発期間と開発総工数のトレードオフを予測することが重要となる [2]．

我々は，作業を並行して行った場合に開発期間と開発総工数に与える影響を予測するためのプロセスモデルと，それを用いたシミュレータを試作している [3]．本稿では，試作したモデルの改良を目的として，実際のプログラムの開発を作業を並行化して行なった場合と並行化せずに行なった場合で，進捗の変化にどのような違いがあるかを調べる実験を行なった．

2 試作シミュレータにおける進捗モデル

2.1 プロセスモデル

図 1に我々の提案するプロセスシミュレーションモデルの概略を示す．本モデルでは，開発プロセスの諸要素を作業，スタッフ，作業間の前後関係，スタッフの割当，の 4 つの要素に集約した上で，作業，スタッフに対して様々な属性を付加する (表 1)．作業の属性である作業の開始時刻を他の作業の終了時刻より先に設定することで，並行作業による開発プロセスを記述することができる．

作業 a の属性	
進捗	$0 \leq p(t) \leq 1$
開始条件	$t = T_s$ または $q(t) = P_s$
	$q(t)$ は上流作業の進捗
終了条件	$t = T_e$ または $p(t) = P_e$
規模	L
難易度	$D > 0$

スタッフ s の属性	
能力	C

表 1: 作業・スタッフの属性

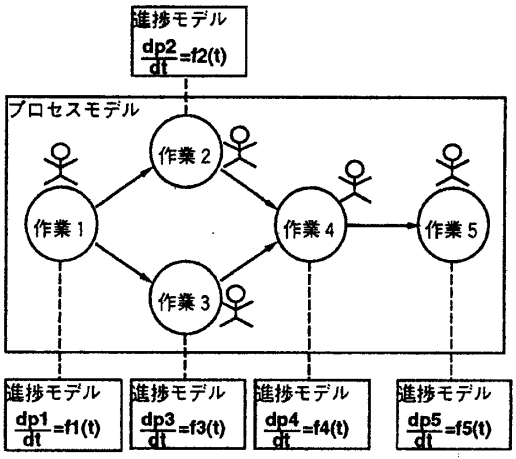

図 1: モデルの概略

2.2 進捗モデル

各作業は，上流作業および下流作業の影響を受け，進捗がどのよう変化するか計算するための「進捗モデル」を持つ．進捗モデルでは作業の進捗を計算する際に前後関係のある作業の進捗をパラメータとすることで，並行作業間の影響を表現することができる．ここでは進捗の変化に次のような仮定をおく．

- 進捗の増加速度は，割り当てられたスタッフの能力，作業の規模，難易度によって変化する．
- 進捗の増加速度は，上流作業の進捗と自作業の進捗の差に比例する
- 下流作業の進捗の影響を受けない

これらの仮定を満たす進捗モデルは形式的には次のような 1 次常微分方程式で記述することができる．

$$\frac{dp}{dt} = \begin{cases} f(t) = K(q(t) - p(t)) & \text{(作業実行中)} \\ 0 & \text{(作業非実行中)} \end{cases}$$

ここで，

$p(t)$: 自作業の進捗
$q(t)$: 上流作業の進捗
K : 進捗速度係数

Kは自作業に割り当てられたスタッフの能力 C，自作業の規模 L，難易度 D によって決まる値である．

2.3 シミュレータ

上記モデルに基づき，開発プロセスの進捗をシミュレートするシステムを試作した．

An experiment on interaction between concurrent activities in software process
Satoshi Yabe, Hajimu Iida, Ken'ichi Matumoto and Koji Torii
†Graduate school of information science, NAIST
‡Information technology center, NAIST

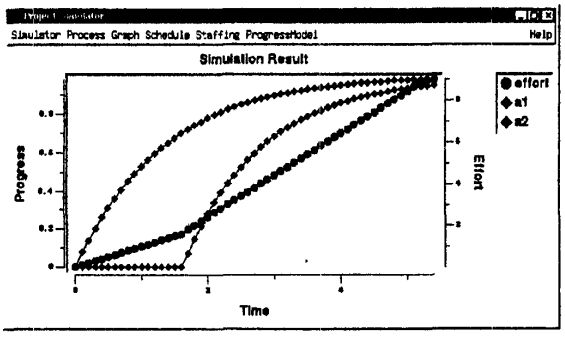

図 2: シミュレーションの例

図 2に二つの作業 (a1, a2) が並行して行われた場合のシミュレータの実行画面を示す．横軸が時間，縦軸が進捗率を表し，effort は累積工数の変化を表している．

3 実験

本シミュレータが用いているモデルが妥当であるかどうかを調べるため，実際に被験者にプログラムを組んでもらい，作業の進捗などを調べる実験を行なった．

実験では，数百行程度の C 言語によるプログラムが得られるような仕様を被験者 (2 名 1 チーム) に与え，設計およびコーディングを行なうというものである．

設計作業とコーディング作業にそれぞれ一人づつ被験者を割り当て，(1) 並行作業による開発と，(2) 設計が終了してからコーディングを行なう開発の 2 回について，被験者を変えて行なった．並行開発時，設計ドキュメントはオンラインで常にアップデートされ，コーディング作業者が参照できるようにした．この時作成した設計ドキュメントを (2) で使用した．

測定したデータは作業にかかった時間と，作業中のキーストローク数である．成果物の途中の行数などではなく，キーストローク数を用いたのは作業が進むに連れて単調に増加するという点で進捗を表すのに向いていると考えられるからである．

ある時刻までのキーストローク数を $s(t)$ とすると，進捗率 $p(t)$ は

$$p(t) = \frac{s(t)}{s(T_e)} \quad (T_e = \text{作業終了時刻})$$

で近似できると考えられる．

また，作業者間のコミュニケーションについても調べるため，並行作業の場合は被験者をビデオで撮影し，会話データなども記録した．

4 実験結果

作業時間とキーストローク数の関係を図 3に示す．横軸が時間 (時間)，縦軸がそれぞれの作業の進捗を示している．今回の実験では上流，下流作業間の影響を調べることを目的としたため，進捗速度係数 Kについては評価の対象外である．したがってコーディング (1) と (2) の開発期間の差自体には意味はない．

4.1 データの分析

実験結果より，次のようなことがわかった．

(1) 前の段階の作業の進み具合が遅くなると，それにつられるようにして，下流作業も遅くなる現象が確認された．

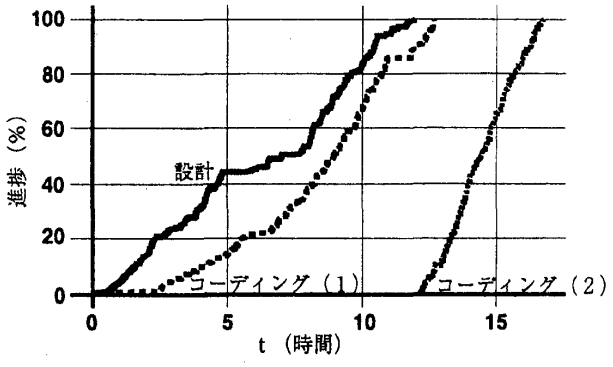

図 3: 実験結果

(2) 上流作業が完全に終ってから下流作業を始めた場合は，キーストローク数はほぼ一定の割合で増加した．

(3) 実験中に撮影したビデオからは，下流作業からの影響として，上流作業の成果物の内容についての苦情・要望・質問や，下流作業が上流に追いついてしまった時の催促などがみられた．

4.2 考察

実験の結果より，上流作業の遅れが下流作業の遅れにつながるという仮定が妥当であることを確かめることができた．しかし，現実には下流作業から上流作業への影響が非常に大きく，シミュレーションモデルにも何らかの形でそれを組み込む必要があことも確認された．

現バージョンの進捗モデルでは，上流から下流への影響が進捗の差に比例すると仮定しているが，これに加えて，下流との差をもとに上流の進捗速度を変化させるよう，モデルの改良を行なうことが考えられる．このためにはコーディングと全く切り離して設計作業を行なった (つまり下流からの影響を除いた) 場合との比較を行なう必要があろう．この点に関しては追加実験を行なう予定である。

5 まとめ

作業を並行して行った場合に開発期間と開発総工数に与える影響を予測するための方法に対して評価を行なった．

今後の課題としては，実験を通して，下流作業からの影響を考慮にいれた進捗モデルの改良と検証ががあげられる．

参考文献

[1] 内藤，飯田，松本，鳥居: “役割別工数投入計画のための見積りモデルの提案”, 情報処理学会研究報告, 96-SE-107, pp.1–8 (1996).

[2] DeMarco T.: “Controlling software projects,” Yourdon Inc. (1982), “ソフトウェア開発プロジェクト技法”, 渡辺 純一訳, 近代科学社 (1987).

[3] 永島，飯田，松本，鳥居: “作業の並行化による影響を考慮した開発プロセスシミュレーションモデルの提案”, 情報処理学会研究報告, 96-SE-109, pp.33–40 (1996).