



Title	プロセスモデル化の例題の協調開発支援管理システムによる実行
Author(s)	三村, 圭一; 飯田, 元; 井上, 克郎 他
Citation	情報処理学会第45回全国大会講演論文集. 1992, 5, p. 157-158
Version Type	VoR
URL	https://hdl.handle.net/11094/50592
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

プロセスモデル化の例題の協調開発支援管理システムによる実行

5S-10

三村圭一[†] 飯田元[†] 井上克郎[†] 鳥居宏次[†][†]大阪大学 [‡]奈良先端科学技術大学院大学

1 はじめに

ソフトウェアの規模と需要の増大につれ、開発効率と信頼性の向上がソフトウェア工学における重要な目標のひとつとなっている。我々は、ソフトウェア開発の効率的な支援を目的として、開発作業の形式化やそれに基づく作業の誘導や自動化に関する研究を行っており、これまでに、複数人によるソフトウェア開発過程の作業モデルを提案し、そのモデルに基づいた支援や管理を行なうシステム「はこにわ」を試作した。

本研究では、はこにわシステムを用いた開発を例題を用いて試験的にを行い、モニタリングによるデータの収集を試みた。

2 はこにわシステムの概要

2.1 作業モデル

はこにわシステムでは、複数人による開発プロセスを「作業モデル」と呼ぶ枠組で表現し、それに基づいた支援・管理を行なう。作業モデル(図1)は、並列に実行される複数の「タスク」によって構成される。タスクは基本作業の集合と、それを用いて定義される文法(ここでは正規文法に制限する)によって構成される。従って、各タスクは、それぞれの持つ基本作業をアルファベットとした正規表現で表すことができる。このとき、基本作業とは、対象となる開発過程をとらえる上での最小の作業のことである。

一つのタスクは、比較的単純な順序で行なわれる作業群を表す。複雑な作業順序は、タスク間のコミュニケーションによって実現される。このため、タスク間では簡単な非同期の通信を行なえることとする。

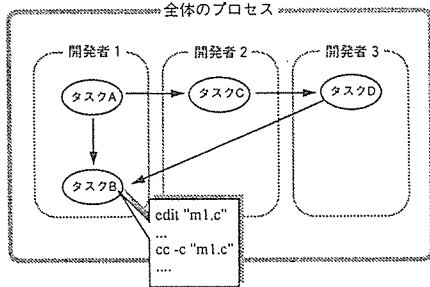


図1: 作業モデルの概要

2.2 はこにわシステム

システムの構成を図2に示す。作業モデルに基づいた開発プロセスの記述から(a)タスクのモニタリングと、タスク間のコミュニケーションのための支援環境(タスクドライバとタスクマネージャ)が生成される。

このシステムでは次の三つの支援を行なう。

(1) 作業の誘導

実際の開発では、複数のタスクを一人の開発者が担当する。そこで、まず管理者がタスクの割当を行ない、その情報を元に、各開発者が用いるための支援インターフェース(タスクドライバ/マネージャ)が生成される。このインターフェースは、タスクごとに定義されている文法に基づいて、各タスクにおける作業の進行をメニューを用いて誘導する。メニューを通じて選択された作業のうち、開発ツールを用いるものなどについては自動的にツールの起動を行なう。

(2) モニタリング

一方で、開発者用インターフェースはメニュー選択によって進行して行く作業の履歴をはこにわサーバに通知する。管理者はこの情報をプロジェクト全体の状態の把握に役立てることができる。

(3) コミュニケーション支援

タスク間のコミュニケーションは、すべてはこにわサーバを介して行なわれ、作業の開始要請や、終了通知などといった、開発者間での単純なコミュニケーションは自動的に行なわれ、記録される。

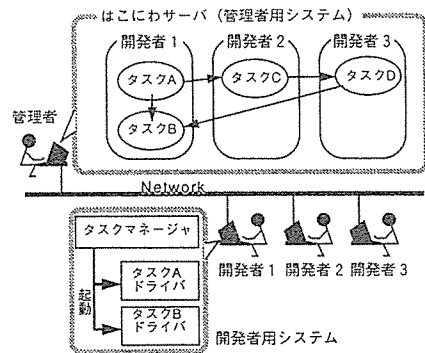


図2: 支援システムの概要

3 運用実験

3.1 例題

はこにわサーバの運用実験として、例題を用いたプログラムの作成を行なった。

例題には酒屋問題[1]を用いた。また、実験で用いた開発プロセスの原型には、Kellnerらによって提案されているプロセスモデリングのための例題[2]を用いた。この問題は、あるソフトウェアシステムの一つのモジュールに対して変更を加える作業を規定しており、8つのサブステップに分けられ、それぞれ様々な開始・終了条件などが付加されている。実験で用いたプロセスは、この例題を元に変更を加えたものである。主な変更点は次のようなものである。

- プログラムはモジュールに分割され、各モジュールは個別に作成される。

ここでは、開発作業を次の3つに大別した。これら3つの作業は直列な関係にある。

作業1:設計及びコーディング

作業2:シンタックスエラーの除去

作業3:テスト及びデバッグ

作業時間はそれぞれの作業に対応するコマンドの実行結果より学生毎に算出した。それぞれの作業時間の分布とそれらを合計して求めた開発期間の分布を図4に示す。

各作業時間及び開発期間の統計量を表1にまとめる。表1中の積率歪度は歪みを表し、時間分布が正(右方向)に歪んでいる時、正の値となる。また、積率尖度は尖りを表し、正規分布では値が3になる。

表1 作業時間の統計値

作業	平均	標準偏差	積率歪度	積率尖度
作業1	8.44	3.26	0.11	2.55
作業2	0.89	1.30	1.98	7.12
作業3	3.25	1.75	0.77	3.23
開発期間	12.58	4.12	-0.11	2.20 (単位: 日)

表1より各作業時間の積率歪度はいずれも正であることがわかる。従って、作業時間分布を表す分布関数としてポアソン分布が適当ではないかと考えられる。また、表1より作業3の積率尖度が3に近い。従って、作業時間分布を表す分布関数として正規分布も考慮する必要があると思われる。

3.2 作業時間の独立性

作業時間分布から開発期間を推定するためには、各作業時間が独立であると仮定する必要がある。この作業時間の独立性を確認するために、作業時間の相関を調べた。作業1と作業3の作業時間の散布図を図5に示す。相関係数は作業1と作業2の間で0.17、作業2と作業3の間で0.04、作業1と作業3の間で0.02である。

従って、本実験における3つの作業時間は互いに独立であると考えられる。

3.3 4つの分布関数との適合度

作業時間分布として三点見積り法では β 分布が、数値シミュレーション法では三角分布がそれぞれ用いられている。これに対し、3.1で示したように、実データの形状からは、ポアソン分布や正規分布が作業時間分布を表す分布関数として適しているように思われる。そこで実データに対するこれら4つの分布関数の適合度を比較する。 χ^2 検定による検定値と検定結果を表2にまとめる。検定の有意水準は10%である。

表2 作業時間分布の検定値と検定結果

分布関数	作業1 χ^2 値(自由度)	作業2 χ^2 値(自由度)	作業3 χ^2 値(自由度)
三角分布	3.98 (8) 採択	15.88 (5) 棄却	2.95 (5) 採択
β 分布	2.90 (8) 採択	6.76 (3) 棄却	4.85 (5) 採択
ポアソン分布	1.89 (7) 採択	3.70 (3) 採択	1.06 (6) 採択
正規分布	2.20 (7) 採択	24.90 (4) 棄却	8.13 (6) 採択

表2より、 β 分布、三角分布、及び、正規分布は作業2で棄却され、ポアソン分布は全ての作業で採択されたことがわかる。また、ポアソン分布では全ての作業で χ^2 値が最小となっていることがわかる。

以上の結果より、プログラム開発の作業時間分布としては、従来用いられていた β 分布や三角分布よりもポアソン分布を使用するのが適していると考えられる。

3.4 ポアソン分布を用いた開発期間の推定

2.3で述べた方法を用い、各作業時間分布にポアソン分布を使用して開発期間に要する時間の分布を求めた。求めた分布と実際の開発期間との間で χ^2 検定を有意水準10%で行なった結果、 χ^2 値7.04、

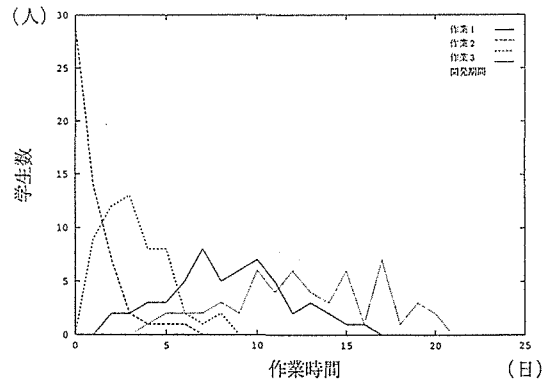


図4 作業時間分布

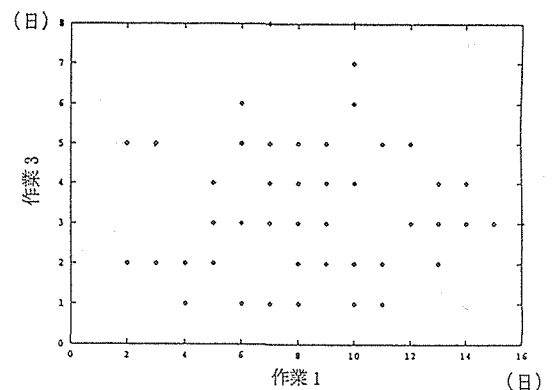


図5 作業時間の散布図

自由度8となり、採択された。

4. あとがき

本稿では大学の学生実験から収集したデータに基づいて、ソフトウェア開発の各作業時間の独立性、及び、作業時間分布と代表的な4つの確率分布との間の適合度を分析した。その結果、各作業時間が独立であることを確認した。更に、作業時間分布を表す分布関数としてポアソン分布を使用することが有効であることを確認した。

現在、作業間の関係がより複雑な開発過程を対象としたデータの収集、分析を検討中である⁽⁴⁾。

文 献

- (1) A. Ralston and E. D. Reilly, Jr.: "Encyclopedia of Computer Science and Engineering", Van Nostrand Reinhold Company(1983), 横上昭男 監訳, "コンピュータ大辞典", pp. 1006-1009, 朝倉書店(1987).
- (2) 情報システムハンドブック編集委員会: "情報システムハンドブック", 培風館(1989).
- (3) 山田茂: "ソフトウェアマネジメントモデル入門", bit, 23, 3, pp. 300-307(1991).
- (4) 渡辺淳志: "作業時間の確率分布を用いたソフトウェア開発期間推定の試み", 大阪大学基礎工学部特別研究報告(1992).