

## 4K-4

## カスタマイズ性の良い業務記述手法の評価\*

坂本泰久 馬島宗平†

NTT ソフトウェア研究所‡

## 1 概要

ソフトウェア開発・保守・運用業務は多くの人間やシステムがからみあった複雑な共同作業である。近年の計算機ネットワーク技術の進歩により従来機能別に構築していた支援システムを統合化し、業務全体の流れを一貫してガイドすることが技術的に可能となってきた。このような統合支援システムの実現のためには、業務全体の仕様を正確に記述するための業務モデルが不可欠である。

業務は技術・組織・法律などの外部環境の変化に常に適応しなければならない性質のものであるから、初めに設定した仕様に対して段階的・付加的な改善要求が多く発生する。そしてこの改善が全体の動作に支障を与えないことの検証は、人手で行なわれるため考慮もれを招きやすい。

そこでこの問題を解決するため、インクリメンタルな手順の記述と実行可能性に関する検証の計算機支援が可能な業務モデルを提案した。そして小規模な作業実験によりこの効果を確認するとともに、モデルに対するその他の要求条件を明らかにした。

## 2 業務モデル

業務には、仕事の割り当てのように「作業中に刻々と変化するリソースの状態に常に適応して決定される側面(動的な側面)」と仕事の手順のように「作業中の動的な変化とは独立に設定すべき側面(静的な側面)」がある。

業務は「役割」という機能別に分けられたプロセスの協力により遂行されているという考え方を前提として「役割・リソース・時間の3要素を軸とした空間内に業務全体を表現するモデル」(3次元業務モデル)を提案した[1]。役割-時間面(業務手順面)が静的な側面に、役割-リソース面(リソース割り付け面)が動的な側面に対応する。両者を合わせたものが業務仕様となる。

業務手順面に関しては、役割を縦軸としたシーケンスチャートで記述することにより、SDE[2]のサービス検証技術が適用できるという仮説を得た[1]。これにより手順のインクリメンタルな記述と実行可能性(業務がどこおりに終了すること)の計算機による検証が可能となる。

リソース割り付け面に関しては、人間やシステムの状態・能力、その時点までの割り付け結果などパラメタが非常に多く、その全てを考慮した割り付けアルゴリズムを手続き的に表現するのは困難である。したがってプロダクションルールなどの宣言的表現により満たすべき条件(制約条件)を並列的に記

\*Evaluating a business model to adopt various changes in environment

†Yasuhisa Sakamoto and Sohei Majima

‡NTT Software Laboratories

述し、それらをすべて満たすように割り付け先を決定する方式が自然だと考えられる。

## 3 評価実験

モデルの有効性を評価するための実験を行った。リソース割り付け面に関しては記法の定義が不十分なため、業務手順面についてのみを評価の対象とした。実験用タスクとして「図書室での文献検索作業」を選択した。これは作業の並行性・柔軟性・トランザクションの多重性などの点で、ソフトウェア開発・保守・運用業務との共通点が多かったためである。

## 3.1 実験方法

作業における役割を管理・判断・検索・コピーの4つに分けた。そして基本の手順と追加する手順を次のように設定した。

基本手順 「複数の文献リスト上の文献を順番に検索し、見つかった文献についてはコピーなどの後処理を行なう」

追加手順 「見つかった文献の参考文献も検索の対象とする」

上記の作業に関して表1に示す2種類の実験を行い、表2の項目について評価した。

表1・実験内容

A. 記述実験	モデルに従って初期手順をシーケンスチャートで記述しサービス検証を行う。次に追加手順を記述し前の仕様と合成する。
B. 作業実験	実験Aで記述したチャートにもとづいて実際に作業を行う。作業は2回行ない、2回目は追加後のチャートで行う。

実験Aは報告者が行なった。シーケンスチャート記述、変換、検証、修正を繰り返しその履歴を記録した。

実験Bは、記法に関する知識を持たない4人を被験者として加え計5人で行った。理解性を評価するため手順の説明はチャートを見せるだけとし口頭での補足などは行わなかった。リソース方向の影響をできるだけ排除するために1人の人間に1つの役割を対応させた。ただし検索担当は大変なので2人に割り当てた。作業中に起こったトラブル(障害・誤り・質問など)は全て時刻とともに記録した。

表2・評価項目

評価項目	内容
1. 記述性	a. インクリメンタルな記述ができるか b. 簡潔に記述できるか c. 必要なことを全て記述できるか
2. 検証性	a. 実行可能性が検証できるか b. 他に検証すべきことがあるか
3. 実用性	a. 人間にとって理解しやすいか

3.2 実験結果

実験 A で記述した作業手順の一部を図 1 に示す。

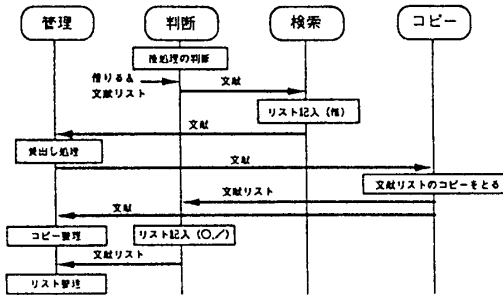


図 1・作業手順チャートの例

実験 B で起こったトラブルの原因別の分類を表 3 に示す。

表 3・発生したトラブル数

評価項目	原因	手順面	分担面	内容
2. 検証性	実行不可能	0	1 (1/0)	0
	仕様不足	2 (1/1)	0	7 (3/4)
3. 実用性	理解不足	13 (12/1)	0	2 (1/1)
	最適化	3 (3/0)	3 (2/1)	1 (0/1)
合計		18 (16/2)	4 (3/1)	10 (4/6)

※ ( ) は 1 回目と 2 回目の内訳

表 3 のうち手順の理解不足によるトラブルの発生時刻を記録したグラフを図 2 に示す。文献リストごとの処理時間が下段に示されている。(2 回目は 1 件しかなかったため省略)

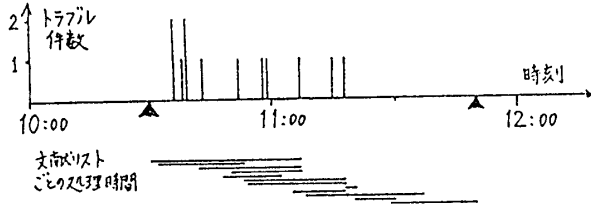


図 2・手順の理解不足によるトラブル発生時刻 (1 回目)

3.3 評価

実験 A における記述の履歴と実験 B におけるトラブルの分析から表 4 の評価結果が得られた。項目欄の記号は表 2 と対応している。評価欄はそれぞれ実験 A, B による評価を示している (○: 問題なし, △: 人間側の問題, ×: 対策が必要)。

表 4・項目別評価結果

項目	評価		内容
	A	B	
1-a	○		追加仕様の独立な記述が可能
1-b	×		処理の重ね合わせの記述が冗長になる
		×	処理の順序が特定されない場合の記述量大
1-c		×	ジョブの優先度に関する記述が必要
2-a	○		繰り返し処理に関するエラーの検出
		○	手順が実行不可能というトラブルは発生せず
2-b	×		データの処理順序に関する意味検証
		△	仕様が正しくても人間的要因により作業しにくい場合がある
		×	複数ジョブの競合による障害の検証
3-a		△	開始後 1 時間程度で理解できるようになった

3.4 考察

記述性について (1-c)

チャートで記述された処理手順はオブジェクト指向におけるクラスとみなすことができる。作業中には入力 (今回の場合は文献リスト) の数だけインスタンスとしてのジョブが発生する。現在のモデルではジョブ間の順序は記述できないが、現実の業務ではジョブに優先順位のある場合が多いので、優先度に関する記述ができることはモデルの要求条件の 1 つになる。例えばソフトウェアの問題処理業務では致命的なバグには最優先で対処しなければならない。

実行可能性以外の検証項目について (2-b)

データ処理順序に関する意味検証は SDE[2] では扱っていないが、ソフトウェア開発のように生産物のある業務に適用する場合には重要な検証ポイントとなる。

また今回の実験では仕様不足による作業時のトラブルが 2 件あったが [表 3]、いずれもジョブ間の競合によるものであった。例えば 1 度検索した文献を再び検索しようとした時に、まだ前のジョブでコピー中だったために検索できないという状況が起こった。この問題を事前に検出できると良いが、そのためには文献というデータに対する排他関係の検証が前提となる。

理解性について (3-a)

作業開始後しばらくの間は手順の理解不足によるトラブルが多かった。これはチャートによる記述を見るのが初めてだったのが大きな原因だと考えられる。仕事が一順したところからはトラブルが減少し [図 2]、仕様を追加した 2 回目もトラブルはほとんどなかった [表 3]。チャートの理解性に関しては、この程度の複雑さの作業において 1 時間程度でチャートにもとづく作業が支障なく行えるようになることが確認された。

リソース割り付けについて

実験では役割を単純に割り振ったが担当者による自発的な判断があり、結果として 3 件の最適化がなされた [表 3 の分担面]。2 章に述べたように割り付けに関しては、リソースの状態に応じた割り付けルールを制約条件として宣言的に記述する方針だが、その実行機構はこのような担当者レベルの最適化要求を吸収できることが望ましい。

4 展望

業務仕様が常に人間によって改善され続けることを前提として、サービス検証技術により人間の負担を軽減する業務記述手法を提案し評価した。今後はリソース割り付けに関する具体的な記法および検証手法を検討する。将来的には業務に携わる全てのメンバから改善要求を吸収し、仕様を調整する技術に発展させていきたいと考えている。

参考文献

[1] 坂本, 馬島: 「カスタマイズ性の良い業務記述手法の検討」, 情報学会第 43 回全国大会 7J-3 Oct.(1991)  
 [2] 加藤など: 「通信サービスの拡大を支援するソフトウェア作成環境 (SDE)」, NTT R&D, 38-11, (1989)